AUTOMATIC EXTRACTION OF ARABIC SUB-WORD UNITS FOR CONTINUOUS SPEECH RECOGNITION

BY

KHALID MOHAMED OQLAH NAHAR

A Dissertation Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

MAY 2013

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This dissertation, written by **KHALID MOHAMED OQLAH NAHAR** under the direction of his dissertation advisor and approved by his dissertation committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILISOPHY IN COMPUTER SCIENCE AND ENGINEERING.**

Prof. Mustafa Elshafei
(Advisor)

3, 17

Dr. Umar Al-Turki
Department Chairman

Dr. Wasfi G. Al-Khatib
(Co-Advisor)

Dr. Salam A. Zummo
Dean of Graduate Studies

13

Dr. Husni Al-Muhtaseb
(Member)

7/7/19

Date

Prof. Radwan Abdel-Aal

(Member)

Dr. Mansour Alghamdi
(Member)

Dedicated to My: Parents, Wife, Children, Sister, Advisor and to My Islamic Ummah

ACKNOWLEDGMENTS

The first acknowledgment is due to my university the King Fahd University of Petroleum & Minerals for supporting this research.

My deep respect, affection, love, loyalty and gratitude for my mother and father up to end of my life. No words could cover what I would like to say to them in this acknowledgment but, may Allah extend their life so I can give them the tenderness and affection they deserve.

I cannot express my feelings towards those who supported me in my study all these years, specifically my wife and my dear children. I would like to thank them for their stolen time and sacrifice. It is hard to forget my dear sister who strongly supported me during my study.

I would like to deeply thank my distinctive advisor Prof. Mustafa Elshafei and say to him that, I will not forget your white hands. My great advisor Prof. Elshafei, I learn from you more than what you expect, I will not forget those great moments that I spent near you. I hope from Allah to extend your life so you can give our Islamic Ummah more and more. Great admit and respect for my co-advisor Dr. Wasfi G. Al-Khatib for his time and support in preparing the research papers. I express my deep thanks for the rest of my thesis/dissertation committee Dr. Husni AL-Muhtaseb, Prof. Radwan E. Abdel-Aal, and Dr. Mansour Alghamdi, for their valuable comments on my dissertation work which improved my skills a lot. In particular, I am indebted to Dr. Husni Al-

Muhtaseb for his intensive review and invaluable comments that brought my thesis to its current shape. May Allah preserve them all.

My grateful thanks to King Abdulaziz City for Science and Technology (KACST) for partially supporting this research work under NSTP project research grant # (08-INF100-4).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARIES AND ABBREVEATIONS

ANN	Artificial Neural Networks
ASR	Automatic Speech Recognition
CD-phoneme	Context Dependent Phoneme
CHMM	Continuous Hidden Markov Model
CI	Context Independent
CIIL	Central Institute of Indian Languages
CMU	Carnegie Melon University
DP	Dynamic Programming
DTW	Dynamic time Wrapping
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HTK	Hidden Markov Model Tool Kit
LM	Language Model
LPCC	Linear Predictive Cepstral Coefficients
LVCSR	Large Vocabulary Continuous Speech Recognition
MFCC	Mel Frequency Cepstrum Coefficients
ML	Maximum Likelihood
MLP	Multilayer Perceptrum
MMSE	Mean Square Error
MSA	Modern Standard Arabic
NLP	Natural Language Processing

OOV	Out Of Vocabulary
SCHMM	Semi-Continuous HMM
SVM	Support Vector Machine
SWUs	Sub-Word Units
WER	Word Error Rate

PUBLICATIONS

- ✓ **Khalid M. O Nahar**, Mustafa Elshafei, Wasfi G. Al-Khatib, Husni Al-Muhtaseb, Mansour M. Alghamdi. “*Statistical Analysis of Arabic Phonemes for continuous Speech Recognition*”, International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 01– Issue 02, November 2012.

- ✓ **Khalid M. O Nahar**, Mustafa Elshafei, Wasfi G. Al-Khatib, Husni Al-Muhtaseb, Mansour M. Alghamdi. “*Statistical Analysis of Arabic Phonemes Used in Arabic Speech Recognition*”, 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part I, pp 533-542.

- ✓ **Nahar, Khalid M.O**; Al-Khatib, Wasfi G.; Elshafei, Moustafa; Al-Muhtaseb, Husni; Alghamdi, Mansour M., "Data-driven Arabic phoneme recognition using varying number of HMM states," Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on , vol., no., pp.1,6, 12-14 Feb. 2013 doi: 10.1109/ICCSPA.2013.6487258 (IEEE).

ABSTRACT

Full Name : KHALID MOHAMED OQLAH NAHAR
Thesis Title : AUTOMATIC EXTRACTION OF ARABIC SUB-WORD UNITS FOR CONTINUOUS SPEECH RECOGNITION.
Major Field : COMPUTER SCIENCE AND ENGINEERING
Date of Degree : MAY 2013

Recent research in speech recognition focuses on speaker independent continuous speech recognition. Continuous speech is more challenging because of variability in the words pronunciations, due to dialects, speaker age, gender, emotional status as well as the nearby words.

The acoustic units, used in representation of language words are called phones. Some existing studies investigated the optimality of English phone set, while no study has been done to investigate the optimality of the Arabic phone set, which is currently used in Arabic speech recognition.

In this research, multiple clustering and segmentation techniques were evaluated on the current Arabic phoneme set for the purpose of automatic extraction of Arabic sub-word units. A hybrid HMM/LVQ-ANN recognition methodology for the existing Arabic phones set was also conducted.

Based on data-driven methods, we studied the determination of the most suitable Arabic sub-word units. We derived different sets of Arabic sub-word units with sizes 30, 50, 70, 90 and 150, and used them in generating the respective dictionaries. The set with 70 sub-word units showed the best recognition rates of 79.3% at the sub-word level and 34.08% at the word level. Although the recognition rate at the word level is not

satisfactory, this work is the first such attempt in extracting Arabic sub-word units using a data-driven approach.

ملخـــــص الرســـالة

الاسم الكامل : خالد محمد عقلة نهار

عنوان الرسالة : الاستخلاص الآلي للوحدات الصوتية العربية للتعرف على الكلام المتصل

التخصص : هندسة وعلوم الحاسوب

تاريخ الدرجة : مايو ٢٠١٣

تهتم الأبحاث الحديثة في مجال التعرف على الأصوات بالتعرف على الصوت المتصل المستقل عن المتحدث. إن تمييز الصوت المتصل هو أكثر تحديا من غيره من الاصوات وذلك بسبب التغير الكبير في نطق الكلمات أثناء الكلام المتواصل ويعزى ذلك إلى لهجة وعمر وجنس المتكلم وإلى قرب أو بعد الكلمات عن بعضها.

إن النموذج الصوتي الذي يمثل كلمات لغة ما يسمى بالألفون. وقد وجدت بعض الدراسات التي تحقق في مدى دقة وأمثلية الفونات الخاصة باللغة الإنجليزية، في حين لا يوجد مثل هذه الدراسات للألفونات العربية والمستخدمة حاليا في تمييز الصوت المتصل.

ومن الجدير بالذكر أنه وخلال هذه ألأطروحة تعرضنا للعديد من خوارزميات وطرق التصنيف والتقطيع للبيانات الصوتية وقمنا بتقييمها بهدف الاستفاده منها في اشتقاق الوحدات

الصوتية الأساسية للصوت العربي المتصل. كما قمنا باستحداث آلية مهجنة من طريقتين وهما نموذج ماركوف الخفي والشبكة العصبية الاصطناعية (HMM/LVQ–ANN) لتمييز الألفونات العربية الموضوعة من قبل خبراء اللغة.

بالاعتماد على طرق الاستخلاص المباشر من البيانات الصوتية نفسها ، حيث قمنا بدراسة إستخلاص الوحدات الصوتية الأساسية العربية الأكثر ملائمة للغة. فقد قمنا بانتاج عدة مجموعات من الوحدات الصوتية الاساسية وبأحجام ٣٠، ٥٠، ٧٠، ٩٠ و ١٥٠ وقمنا باستخدام هذه المجموعات في انتاج المدونات الصوتية المقابلة لكل مجموعه.

لقد تبين لنا أن المجموعة المكونة من ٧٠ عنصرا من الوحدات الصوتية الأساسية كانت الأفضل من ناحية الدقة عند استخدامها في تمييز الصوت سواء على مستوى الوحدات الصوتية الأساسية أو على مستوى الكلمات . حيث بلغت نسبة الأولى ٧٩,٣% والثانية ٣٤,٠٨% و بالرغم من أن النسبة على مستوى الكلمات غير مقنعه إلا أن هذه الطريقة تعتبر فريدة من نوعها في مجال استخلاص الوحدات الصوتية الأساسية العربية بطريقة الاستخلاص المباشر من البيانات الصوتية.

CHAPTER 1

INTRODUCTION

"Speech is the most natural form of human communication" (Sharma et al., 2012). Over many decades, scientific academic community tried to tackle the problem of automatic speech recognition. Many successful attempts were done and much more are needed to be done in the future, which makes speech recognition as one of the important topics for research. Automatic Speech Recognition (ASR) is defined as the process of converting audio waves (speech acoustic signals) to its corresponding set of words or other linguistic units based on a specific algorithm (He and Deng, 2008). ASR appears in many applications and Information Technology (IT)-solutions for industrial and civil areas such as: hands free operations, mobile voice application, human-computer interaction, automatic translation, hearing aids for handicap, automatic dictation and simplified man-machine communication (via-voice systems). In ASR, continuous speech recognition is more challenging than isolated-word recognition.

The performance of ASR is highly affected by several conditions in their operations such as: the vocabulary size, speaker dependency and, noisy environment. The recognition performance increased when using small vocabulary and speaker-dependent condition, while using large vocabulary and speaker-independent the decreases performance significantly.

The large vocabulary continuous speech recognition systems do not depend on the whole word for recognition, because of the high number of words that may exist in the vocabulary (Singh et al., 2002). The need to have enough training examples of the words and the word itself may not be recognized during the recognition phase. As a consequence, ASR uses much smaller units than a word; it is called sub-word units (SWUs). We will refer to this set of SWUs as phones, although it could be a mixture, allophones, or any arbitrary sub-words.

An ASR system may consists of two stages; the first stage is the training stage where we build and train the Hidden Markov Models (HMM's), see Figure 1-1. In small vocabulary speech recognition systems, HMM models are trained for each spoken word (Dua et al., 2012).

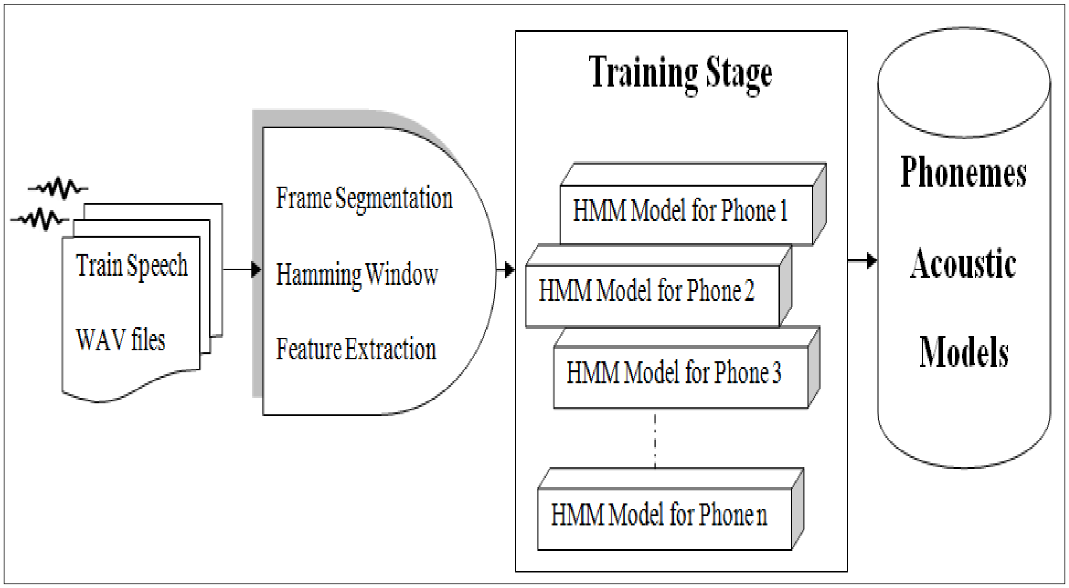


Figure 1-1: Training Stage of Speech Recognition

The second stage in ASR system is the decoding stage, where words models compete to describe the observed speech, the set of words with the highest observed probability (Maximum likelihoods) are to be chosen to represent the words transcription. The language model is used to increase the decoding accuracy and reduce word error rate. One of the basic blocks in any speech recognition system is the acoustic models; these acoustic models need a suitable representation in order to be recognized later (Kenney, 2008).

The most popular representation of speech atomic entities are; tri-phones. successful ASR systems are typically based on one or a combination of the following techniques for modeling tri-phones (Levinson, 1986):

- Hidden Markov Models (HMM)
- Dynamic Time Warping (DTW)
- Neural Networks (NN)
- Hybrid Models (NN-HMM)

Moreover, the typical ASR has another important component which determines the relationship between words and phonemes; it is the dictionary or the lexicon. The dictionary gives all allowable phonemes sequences and pronunciations variations (Kenney, 2008). Figure 1-2 (MITCogNet, 2010) , shows the standard ASR components.

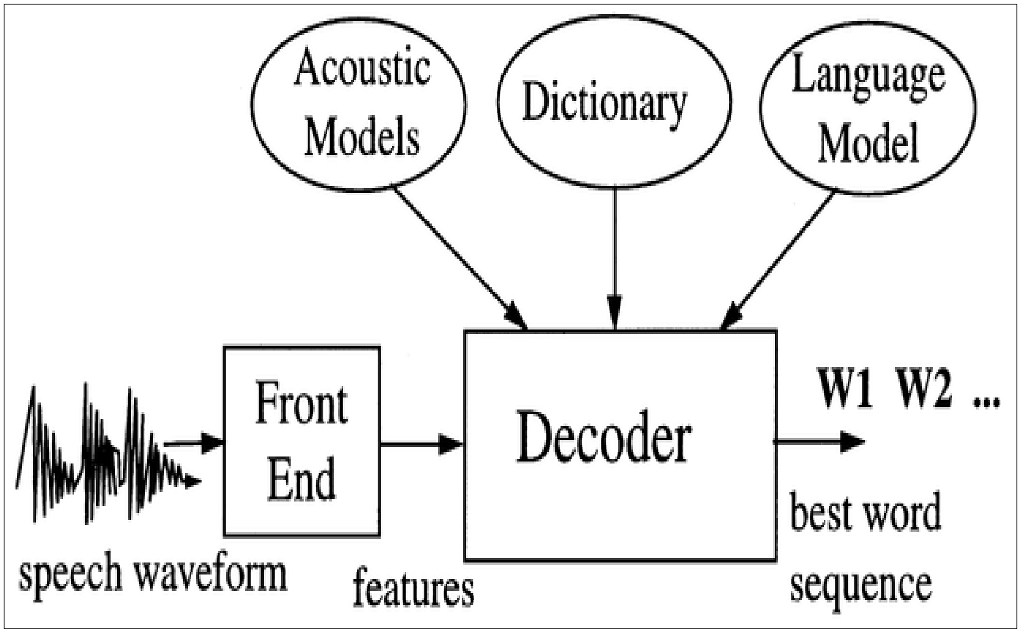


Figure 1-2: A Speech Recognition System (MITCogNet, 2010)

Phonemes recognition in continuous speech is not an easy task due to coarticulation, and the inherent variability in the pronunciation of some phonemes. For example; stop consonants are short duration phonemes that are usually misclassified. State–of-the-art continuous Arabic speech recognition (ASR) systems recognize modern standard Arabic (MSA) that usually appears in TV-News, newspapers, and books. MSA is the formal style of writing and speech across the Arab world (Elmahdy et al., 2011).

Arabic dialects and accents, however, pose great challenges in developing automatic speech recognition systems (ASR). For example, MSA-based ASR systems suffer from high word error rate (WER) when applied to those dialects and accents. As a consequence, a data driven approach needs to be investigated in order to check its ability to recognize and transcribe Arabic phonemes for non MSA that exist in other dialects.

The current available SWUs are defined by phoneticians or expert linguistics. In this research we will focus on continuous Arabic speech recognition and, we will address the selection of the phoneme set or SWUs automatically using a data driven approach.

The Standard ASR system is based on context dependent phonemes as basic SWUs. The acoustic model is trained for each one of the triphone of these phonemes. As we mentioned earlier, the words can't be used in the recognition process therefore, we need a better frequent sub-words that may occur more frequently than the whole word in the corpus, and in this case it is worthy to investigate the sub-word approach which may be the reasonable solution for getting better trained system. These SWUs, if carefully selected, could extend the recognition vocabulary to words that are not recognized during the training phase of the recognition system. We have to keep in mind that the dictionary and the utterances used in the training corpus are fully dependent on the phoneme set defined by expert linguistics. As a result, any misleading boundaries between phonemes or their identity, will lead to bad training for the acoustic models, and increases the word error rate in the recognition phase. Removing the dependency on the pre-defined phonemes and automatically determining the SWUs will reveal or minimize the mismatch between actual realizations and transcriptions in the dictionary, which in turn increases the recognition rate. In Arabic, defining the phoneme set is fully based on linguistic knowledge, skills and experience. It is not derived from the speech data itself, which is considered being a weak point in Arabic speech recognition system. Table 4-1 shows the list of the Arabic phonemes Symbols (Ali, Elshafei, Al-Ghamdi, et al., 2008) currently used in continuous Arabic speech recognition with their meanings.

1.1 Thesis Statement

This thesis investigates and evaluates one parts of the continuous Arabic speech recognition system: the phoneme set which is mainly used in transcribing the utterances and the dictionary as well. Accurate differentiation of the SWUs in the speech recognizer will increase the accuracy of the recognition and reduces the word error rate. The current phoneme set is being put by expert linguistics, and as such is totally knowledge-based. In this thesis, a data driven approach is developed to extract the set of SWUs automatically and evaluate them. The proposed SWUs will be used in continuous Arabic speech recognition in place of the knowledge-based phoneme set.

Evaluation of the recognition accuracy for continuous Arabic speech recognition using the knowledge-based phoneme set and using data-driven SWUs are carried out. Sensitivity analysis is carried out and recognition of the words is presented.

1.2 Thesis Motivation

Many natural language processing (NLP) applications embed the speech recognizer as a front-end. Applications such as via-voice systems that are designed for persons with special needs, applications designed for learning interactively with computer through human-computer interaction, security systems which depend mainly on sound-print, voice dialogue systems, voice mobile applications and some control systems. Improving the speech recognition system will enhance those applications that use speech recognition as a front-end.

State of the art of Arabic speech recognizer accuracy and efficiency highly depend on the set of Arabic phonemes, which is defined by expert linguistics. Evaluating this set of phonemes and converting them from fixed set to a data driven one, will enhance the flexibility and accuracy of the Arabic speech recognizer. In the same context, the possibility and ability to recognize the speech of different accents will increase since the phoneme set is totally data driven and no human intervention is needed.

This thesis evaluates the current phoneme set and proposes data-driven SWUs in order to improve the performance of the Arabic speech recognition.

1.3 Thesis Objectives

The main goal of this thesis is to investigate the automatic extraction of the Arabic SWUs using data driven approach, which may improve and enhance Arabic ASR systems to be able to recognize non-modern standard Arabic. The focus will be on continuous Arabic speech recognition based on SWUs since there is no significant reported research in this area. No linguistic expert involvement is needed to clarify the acoustic SWUs because sub-words are automatically generated. Consequently, in this research, we address the need for a data driven approach for identifying and transcribing Arabic SWUs. The developed framework can then be used for non-MSA speech. In addition, extending the Arabic ASR through the automatic extraction of the SWUs would increase the flexibility of the Arabic ASR system to handle Arabic accents in the future.

The main objectives of this thesis are divided as follows:

First, the current Arabic ASR system (Schlüter et al., 2008), (Soltau et al., 2007), (Diehl and Gales, 2012) is evaluated which depends on a knowledge-based set of phonemes. A statistical analysis study is to be carried out to enhance the current phonemes recognition. A phoneme level transcription is to be done based on the previous statistical study with and without using the language model (LM), the knowledge-based list of phonemes is used.

Second, we will apply the learning vector quantization (LVQ) algorithm developed by (Kohonen, 1988) to enhance the phoneme recognition rate. We will try both dependent and independent classification techniques.

Third, we will use the K-means classification algorithm to determine a new set of SWUs instead of the knowledge-based phoneme list and we will apply the LVQ and HMM/LVQ algorithms to evaluate the usefulness and accuracy of the newly derived set of SWUs. The work will be done on the frame level and using single state HMM for each sub-word unit.

Fourth, sensitivity analysis will be carried out to compare the recognition of the newly SWUs based Arabic ASR system with the baseline ASR system.

The Hidden Markov Model tool-kit (HTK), Carnegie Mellon University (CMU) Sphinx speech recognition engine and MATLAB will be used to investigate the above cited objectives. The Sphinx engine will be applied on the baseline system, which contains a pronunciation dictionary of 14,234 words from 5.4 hours of Arabic broadcast news corpus and will be used to perform utterance forced alignment. The HTK will be used for phoneme transcription. MATLAB will be used to carry out statistical analysis, K-means classification and applying LVQ.

1.4 Thesis Contributions

The main contribution of this thesis is the new data driven approach used for extracting the SWUs that may be used instead of the knowledge-based phonemes. In addition, an enhancement was achieved in the Arabic SWUs transcription compared to phonemes transcription. Moreover, we investigated the use of the SWUs in Arabic speech recognition and compared it to the baseline system. Our results show the following findings:

A data driven set of SWUs for continuous Arabic speech recognition was generated. SWUs are determined using the K-means clustering technique which was improved using LVQ and evaluated using LVQ/HMM on phoneme level recognition.

A Complete statistical analysis of Knowledge-based Arabic phonemes was carried out, which assisted our evaluation of the extracted SWUs. The analysis includes; frequency, mean length, mode length, bigram table, phoneme clustering and trigram table. These results were utilized in the next finding. The result of this study embeds to determine the most appropriate number of HMM states for each phoneme.

A phoneme transcription methodology that could be applied on the extracted SWUs has been designed. The same phoneme transcription methodology was carried out on frame level and using single state HMMs.

Successful Hybrid LVQ/HMM-ANN recognition methodology for Arabic phoneme recognition has been suggested. This hybrid methodology has been applied on the extracted SWUs using HTK.

A complete algorithm to generate the newly SWUs based dictionary has been designed. This algorithm can be used later used in continuous speech recognition.

1.5 Thesis Outline

The rest of this thesis consists of the following chapters (see

Figure 1-3). Chapter 2 (THEORETICAL BACKGROUND) outlines the theory behind speech recognition. Chapter 3 (LITERATURE REVIEW) presents the literature review on extracting SWUs.

Chapter 4 (STATISTICAL ANALYSIS OF ARABIC PHONEMES) describes the used corpus and presents a statistical analysis for the existing Arabic language phonemes used in continuous speech recognition. Chapter 5 (ARABIC PHONEMES TRANSCRIPTION USING DATA DRIVEN APPROACH) employs the results of Chapter 4 to improve Arabic phonemes transcription.

Chapter 6 (ARABIC PHONEMES RECOGNITION USING LVQ) investigates the Arabic phonemes recognition using standalone Learning Vector Quantization (LVQ) without using any bigrams model or learning algorithms. Chapter 7 (ARABIC PHONEMES RECOGNITION USING COMBINED LVQ AND HMM MODEL) is a continuation of the work done in Chapter 6 where a combination of LVQ and HMM is used for Arabic phoneme recognition using Viterbi algorithm.

Chapter 8 (DATA-DRIVEN ARABIC SUB-WORD UNITS EXTRACTION USING LVQ) presents extracting the SWUs based on what we have done in previous chapters.

Chapter 9 (SPEECH RECOGNITION USING EXTRACTED SUB-WORD UNITS) shows our work in validating extracted SWUs by using them in continuous Arabic speech recognition. Finally, chapter 10 (CONCLUSION AND FUTURE RESEARCH) concludes our work and gives future research directions.

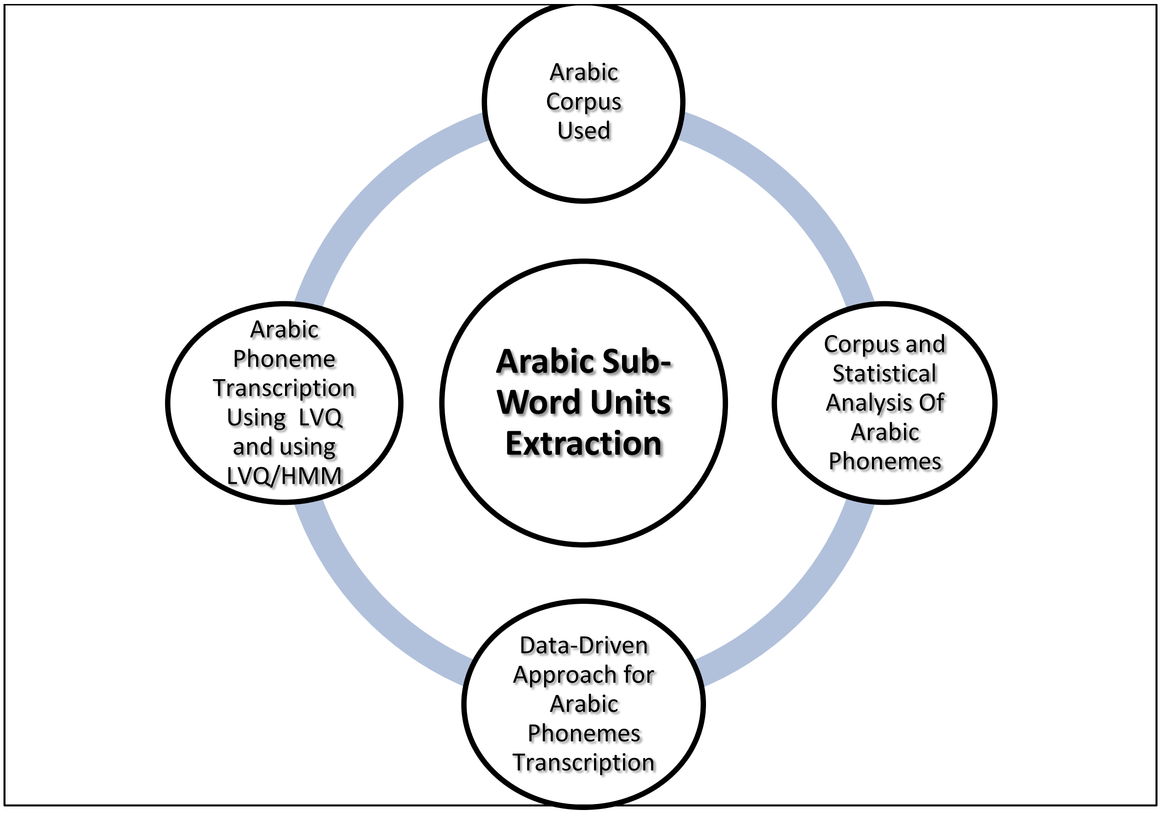

Arabic Corpus Used
Arabic Phoneme Transcription Using LVQ and using LVQ/HMM
Arabic Sub-Word Units Extraction
Corpus and Statistical Analysis Of Arabic Phonemes
Data-Driven Approach for Arabic Phonemes Transcription

Figure 1-3 : SWUs Extraction Road Map

CHAPTER 2

THEORETICAL BACKGROUND

2.1 INTRODUCTION

Speech recognition systems theory is strongly related to a variety of fields including; Mathematical algebra, probability and statistics, information theory, linguistics, stochastic processing and various aspects of computer science. Speech recognition has therefore been viewed as an advanced research topic in computer science and engineering. In this chapter we will cover the theory behind some important models and algorithms used in this thesis including: HMM, Viterbi algorithm, Baum Welch algorithm, LVQ algorithm and, K-means algorithm. Table 2-1, lists the algorithms used during this research and the purpose behind each one. The following sections will briefly address these algorithms.

Table 2-1: The Algorithms and Their Use

Algorithm	Used For
HMM	Learning Model
Viterbi Algorithm	Most Probable Path in a Set of Paths
Baum Welch Algorithm	Used for Training HMMs
LVQ	Efficient Classification Algorithm
K-means	Initial Classification algorithm

2.2 Hidden Markov Model (HMM)

Most of the speech recognition systems depend on the Hidden Markov Models (HMM), it is a statistical model based on the initiating research work of (Baker, 1975). Figure 2-1 shows the common representation of Arabic phonemes using HMM. Arabic phonemes are usually represented by three HMM emitting states.

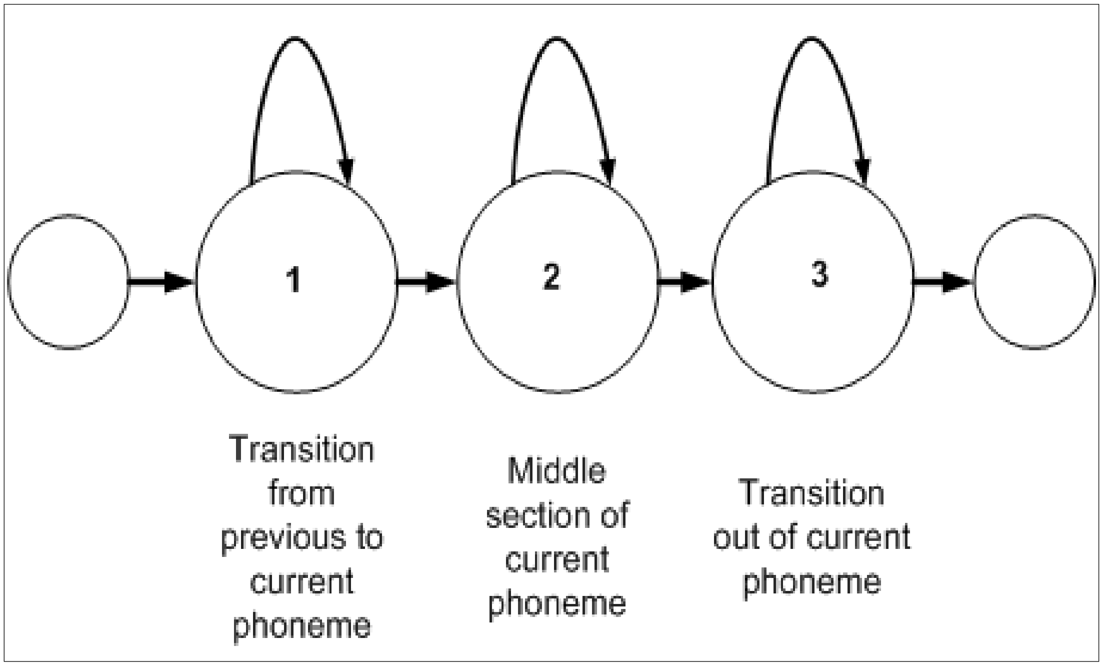

1
2
3
Transition from previous to current phoneme
Middle section of current phoneme
Transition out of current phoneme

Figure 2-1: Standard HMM Model for a Phoneme.

For large vocabulary speaker–independent continuous speech recognition systems, there are two notable successful tools in the research community; the HMM model toolkit (HTK) which was developed at Cambridge University and the Sphinx system developed at Carnegie Mellon University (CMU) (Lee, 1988). HMM is a finite automaton with the probability on the transition arc (Rabiner, 1989), represented by a model $\lambda = (N, \pi, A, B)$ where:

$$\textbf{Number Of States N} \tag{2-1}$$

State transition probabilities **A:**

$$\mathbf{A} = \{\mathbf{a_{ij}}\},\ \ \mathbf{a_{ij}} = \mathbf{p}(\mathbf{s_t} = \mathbf{j} \mid \mathbf{s_{t-1}} = \mathbf{i}),\ \ \mathbf{1} \le \mathbf{i}, \mathbf{j} \le \mathbf{N} \tag{2-2}$$

Where **P** is the probability, (s_t) is the state at time (t)**.**

Vector of initial state probabilities$\boldsymbol{\pi}$**:**

$$\boldsymbol{\pi} = \{\boldsymbol{\pi}_\mathbf{i}\},\ \ \boldsymbol{\pi}_\mathbf{i} = \mathbf{p}(\mathbf{s_1} = \mathbf{i}), \mathbf{1} \le \mathbf{i} \le \mathbf{N} \tag{2-3}$$

Observation probability symbol $\mathbf{B=\{b_j(x_t)\}}$ **where:**

$$\mathbf{b_j}(\mathbf{x_t}) = \mathbf{p}(\mathbf{x_t} \mid \mathbf{s_t} = \mathbf{j}) \tag{2-4}$$

Where (X_t) is the observation at time (t).

Figure 2-2 illustrates the HMM different parameters. The graph represents a discrete HMM model formed of five states where each state has three observation probabilities with the transition probability on the transition arc. Each state in the HMM can generate three different observations according to its output probability density function (PDF). No one-to-one correspondence between the HMM states sequence and the observations sequence, so we cannot determine the state sequence for a given observation sequence; i.e., the state sequence is not observable and therefore hidden. This explained the placement of the word "*hidden"* in the statement *Hidden Markov Model* (Huang et al., 2001). Since a_{ij} , π_i and $b_j(x_t)$ all are probabilities, they must satisfy the following conditions:

$$\mathbf{a_{ij} \geq 0, \quad b_i(k) \geq 0 \text{ and } \pi_i \geq 0 \quad \forall\ i, j, k\ \in [1, N]} \tag{2-5}$$

$$\sum_{\mathbf{j=1}}^{\mathbf{N}} \mathbf{a_{ij}} = \mathbf{1} \tag{2-6}$$

$$\sum_{\mathbf{k=1}}^{\mathbf{N}} \mathbf{b_i(k)} = \mathbf{1} \tag{2-7}$$

$$\sum_{\mathbf{i=1}}^{\mathbf{N}} \boldsymbol{\pi}_{\mathbf{i}} = \mathbf{1} \tag{2-8}$$

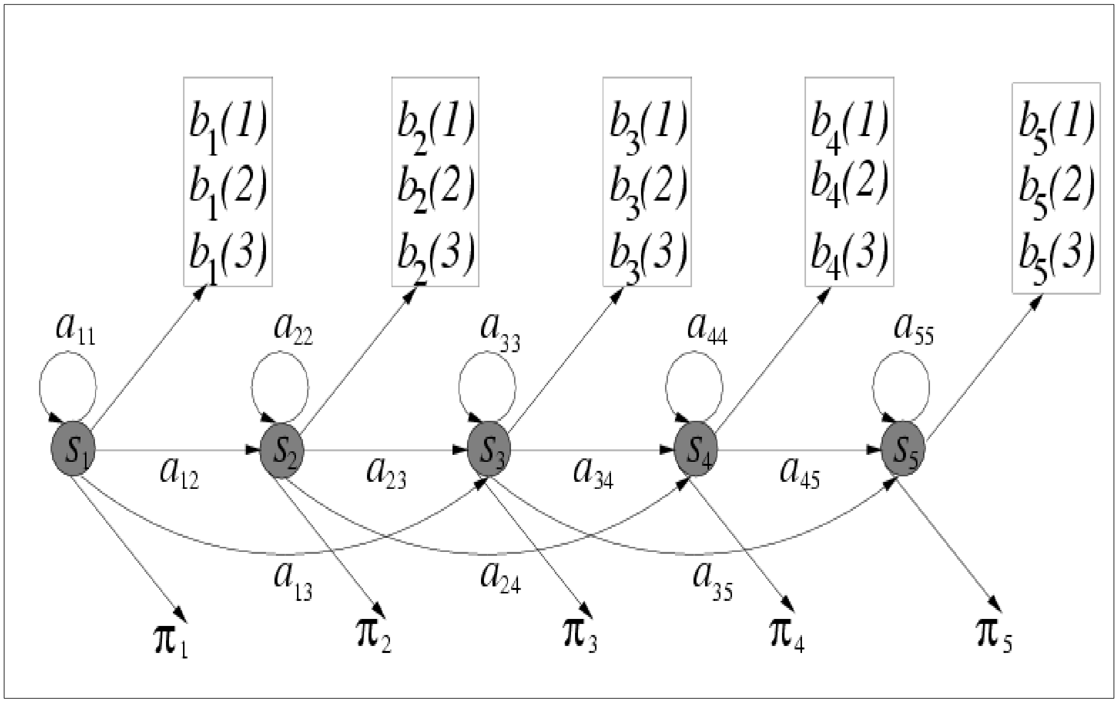


Figure 2-2: HMM Parameters Definition (Schmied, 2008).

2.2.1 HMM Fundamental Problems

HMM design has three fundamental problems; those problems must be solved so that the HMM model becomes useful in real-world applications. These problems are (Rabiner, 1989):

Problem one – Evaluation or Recognition

Given the observation sequence **O** = $(O_1,O_2,...,O_t)$ and the model $\boldsymbol{\lambda}$= (N,π, A,B), how to compute the probability of the observation sequence given the model? That is, how to calculate **P (O|λ)** efficiently? If we consider the case where we have a set of models compete to solve a specific problem, this problem allows us to choose the HMM model which best matches the observations.

Problem two - Optimal or Correct State Sequence (Decoding Problem)

Given the observation sequence **O** = $(O_1,O_2,...,O_t)$ and the model $\boldsymbol{\lambda}$ = (N, π, A, B), how is a corresponding state sequence, **q** = $(q_1, q_2,...,q_t)$, chosen to be optimal in some sense (i.e. best “explains” the observations)?. Unfortunately there is no correct sequence that could be found. Multiple optimal criteria have been applied to uncover the hidden part of the model. Most of the users of the HMM in continuous speech recognition learn about the structure of the model, to find optimal state sequences or to get average statistics of individual states.

Problem three – Adjustment or Optimization (Learning Problem)

How are the probability measures, $\boldsymbol{\lambda}$ = (N,π, A, B), adjusted to maximize **P(O|λ)**? . The observation sequence used to adjust the model parameters is called training sequence since it used to train the HMM. Training is crucial to most applications.

In order to find the most probable sequence of observations and to maximize this probability, two algorithms are used to enhance the recognition system based on HMM; they are Viterbi and Baum Welch algorithms respectively.

2.2.2 HMM Limitations

It should be noted here that although HMM has contributed in advanced speech recognition it has some limitations including (Rabiner, 1989):

The major limitation is the assumption that successive observations (Frames of Speech) are independent i.e.:

$$\mathbf{P}\,(\mathbf{O_1}\ \mathbf{O_2}\ \mathbf{O_3}\ldots\mathbf{O_T}) = \prod_{\mathbf{i=1}}^{\mathbf{T}} \mathbf{P}(\mathbf{O_i}) \qquad (2\text{-}9)$$

The assumption that the distributions of individual observation parameters can be well represented as a mixture of Gaussian or autoregressive densities.

All observation frames are dependent only on the state that generated them, not on the neighboring observation frames i.e. the probability of being at given state at time (t) only depends on the state at time (t-1), which is inappropriate for speech sounds. There is no information about the HMM state duration

Despite the previous limitations, the model works reasonably well in many speech recognition applications.

2.2.3 Solutions to HMM Problems

Solution to Problem 1 – Evaluation or Recognition

Since we wish to calculate the probability of the observation sequence $\mathbf{O}=O_1O_2O_3\ldots O_T$, given the model $\boldsymbol{\lambda} = (N,\pi,A,B)$, i.e. $\mathbf{P(O|\lambda)}$**,** the most straightforward way of doing this is through enumerating every possible state sequence of the length **T** (the number of observations). By considering a set of fixed state sequence $\mathbf{Q}=q_1 q_2q_3\ldots.q_T$, the probability of **O** (given the model) is obtained by summing this joint probability over all possible state sequences **Q** i.e.

$$\mathbf{P(O|\lambda) = \sum\nolimits_{all\ Q} P(O|Q,\lambda)P(Q|\lambda)} \tag{2-10}$$

$$\mathbf{P(O|\lambda) = \sum\nolimits_{q_1q_2..q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1q_2} b_{q_2}(O_2) \ldots\ldots\ldots. a_{q_{T-1}q_T}(Q_T)} \tag{2-11}$$

From implementation point of view an efficient procedure is required to solve problem 1. Fortunately the procedure exist and it is called Forward-Backward Procedure which is the core of the Baum-Welch algorithm (Rabiner, 1989).

Solution to Problem 2-Correct Sequence

In problem1 an exact solution exist, while several possible solutions exist for problem 2 by finding the optimal state sequence associated with the given observation sequence. The difficulty lies in defining the optimal state sequence associated with the observation sequence. One possible criterion is to choose the state "q_t" which is individually most likely at time t, which maximizes the expected number of correct states. When a HMM state has a transition probability equal to zero, the resulting state sequence would be invalid. To solve this problem the most likely double or triple states are taken but, the common way used is to find the best unique state sequence (path). A formal technique for finding this single best path based on dynamic programming is called Viterbi Algorithm (Rabiner, 1989).

Solution to Problem 3 – Adjustment Problem

The most difficult problem of HMM, is to adjust the model parameters (N,π,A,B) to maximize the probability of the observation sequence given the model. There is no analytical way to adjust the parameters. The model $\boldsymbol{\lambda}$ is chosen so that $\mathbf{P\ (O|\boldsymbol{\lambda})}$ is locally maximized. After that, iterative procedures like Baum-Welch algorithm, Expectation Maximization (EM) or gradient technique is used to update the parameters. The most frequently used algorithm to solve this problem is the iterative Baum – Welch algorithm (Rabiner, 1989).

2.3 Forward Algorithm

We use the forward algorithm to calculate the probability of a (T) long observation sequence:

$$\mathbf{X^{(k)} = X_{k1}X_{k2}X_{k3}X_{k5} \dots \dots \dots X_{kT}} \qquad (2\text{-}12)$$

Where, each of the X is in the observable set. The intermediate probabilities α's are calculated by a recursive calculation by initially calculating $\propto$ for all states at t=1 using the following formula (Rabiner, 1989):

$$\boldsymbol{\propto_1}\ (\mathbf{j}) = \boldsymbol{\pi}_\mathbf{j}\mathbf{b_{jk1}} \qquad (2\text{-}13)$$

For the time from 2 up to T , the partial probability $\propto$ is calculated for each state using the following formula (Rabiner, 1989):

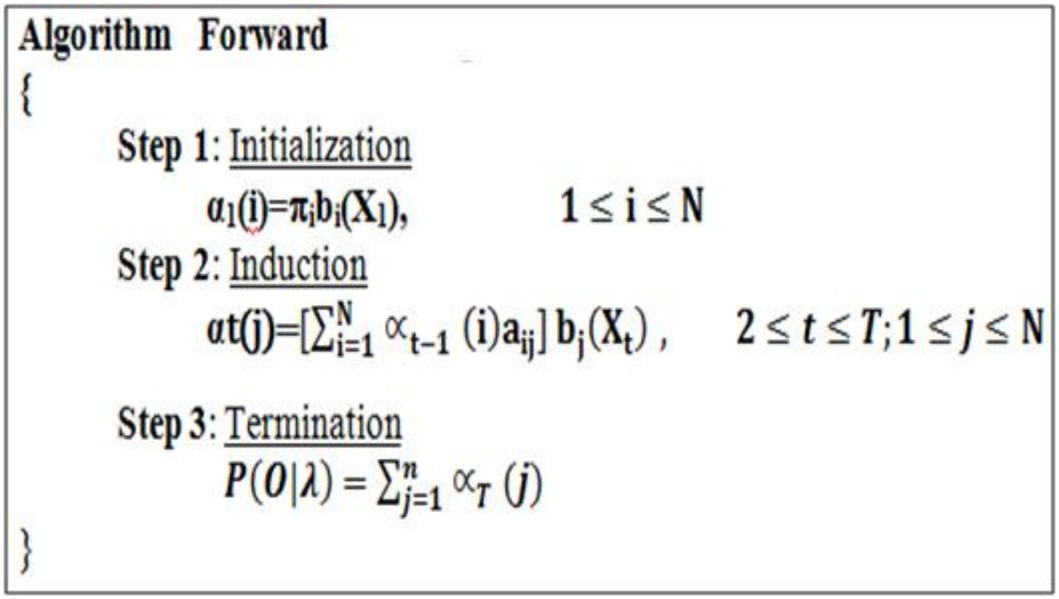
$$\propto_{t+1}(j) = \sum_{i=1}^{n} (\propto_t (i) a_{ij}) b_{j_{k_t}} \qquad (2\text{-}14)$$

Equation (2-14), represents the product of the appropriate observation probability and the sum over all possible routes to that state, exploiting recursion by knowing these values already for the previous time step.

Finally , the sum of all partial probabilities gives the probability of the P(O|λ) by the formula (Rabiner, 1989):

$$P\left(X^{(k)}\right) = \sum_{j=1}^{n} \propto_T (j) \qquad (2\text{-}15)$$

To recap, each partial probability (at time $t > 2$) is calculated from all the previous states. A psedu code for the algorithm is listed in Figure 2-3.



```
Algorithm  Forward
{
    Step 1: Initialization
        $\alpha_1(i)=\pi_i b_i(X_1)$,        $1 \le i \le N$
    Step 2: Induction
        $\alpha t(j)=[\sum_{i=1}^{N} \propto_{t-1} (i)a_{ij}]\, b_j(X_t)$ ,     $2 \le t \le T; 1 \le j \le N$

    Step 3: Termination
        $P(O|\lambda) = \sum_{j=1}^{n} \propto_T (j)$
}
```

Figure 2-3: Forward Algorithm (Huang et al., 2001)

2.4 Baum Welch Algorithm

Baum Welch Algorithm is known also as forward-backward algorithm, it was introduced by (Baum et al., 1970). It is used to train Markov Model without using manually annotated corpora. It is considered as a special case of estimation maximization algorithm. The algorithm assigns initial probabilities to all HMM parameters and trains the HMM's.

When the training process converges, it adjusts the HMM's parameters in order to increase the probability assigned to the training set. When prior knowledge is available about the initial probability it will be assigned to HMM parameters, otherwise a random or flat initialization will be assigned to all HMM's. The algorithm goes in a loop where in every iteration the new probabilities are estimated, which means adjusting the HMM parameters in each iteration. The training will be stopped when no improvement on the probability in successive iterations is being noted.

Baum-Welch tries to reach the local maxima of the probability function $P(O|\lambda)$. The model always converges, but global maximization is not guaranteed.

This recursion depends on the fact that the probability of being in state (j) at time (t) and seeing observation (O_t) can be deduced by summing the forward probabilities for all possible predecessor states (i) weighted by the transition probability (a_{ij}), Figure 2-4 shows the Baum-Welch stages. For more details of the Baum-Welch equations and Derivations we can refer to (Huang et al., 2001) and (Rabiner, 1989).

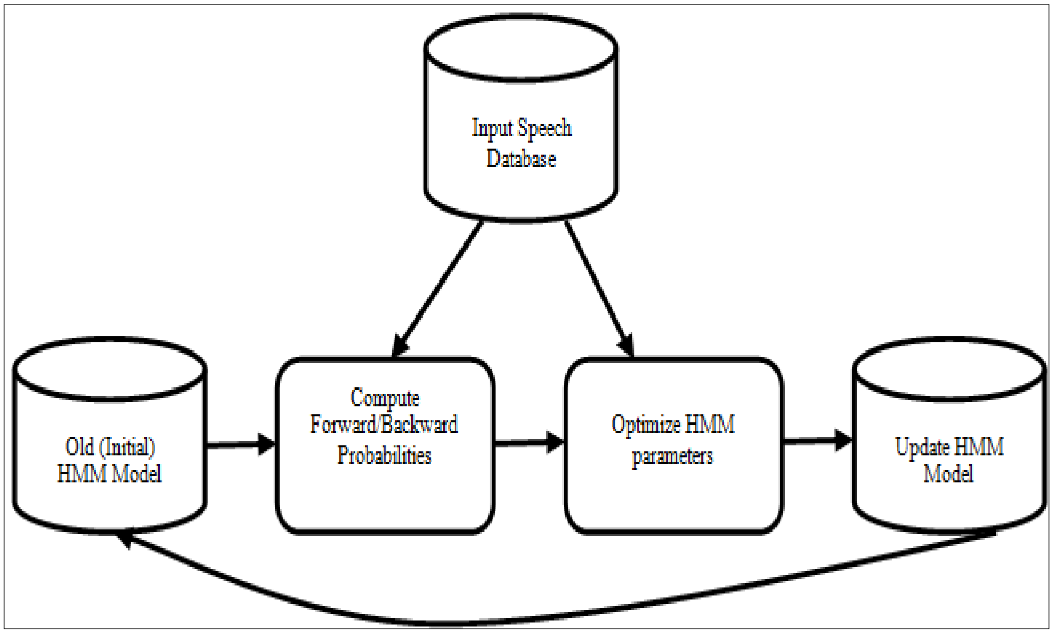


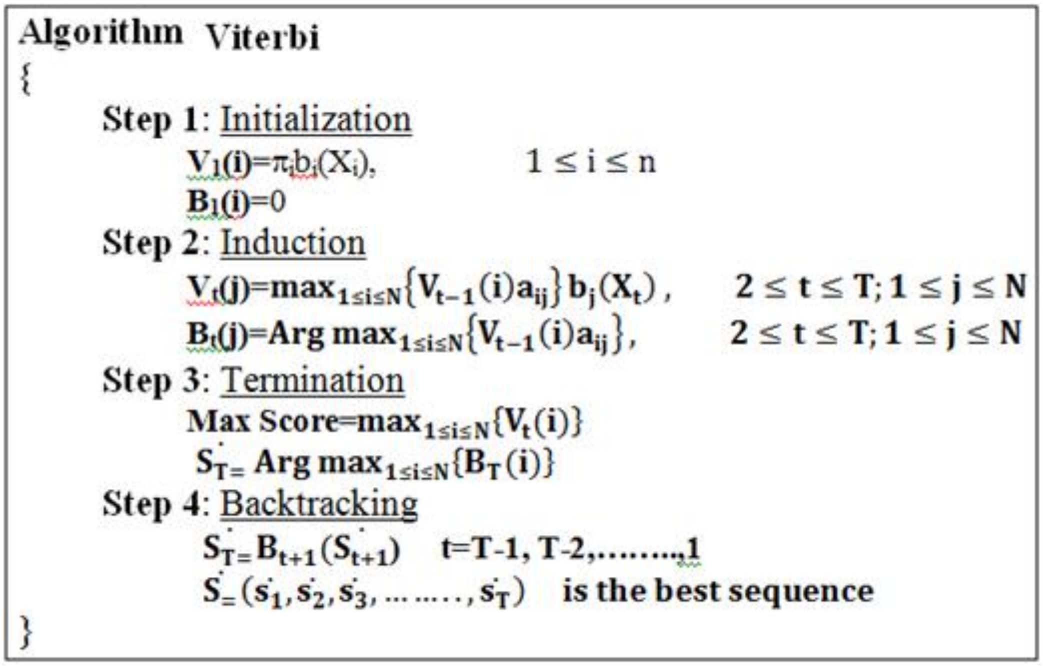
Figure 2-4: Baum Welch Algorithm Stages

2.5 Viterbi Algorithm

For a given HMM, the Viterbi Algorithm (VA) is a recursive optimal solution used to estimate the hidden state sequence of Markov process. The algorithm keeps track of a backward pointer for each state and stores its probability, which indicates the probability of reaching the state through the route that is indicated by the backward pointer. The path with the maximum probability to final state is to be considered. Based on the whole HMM state sequence, the VA can determine the most probable path for a given time sequence events. Based on this property, the VA becomes a core part of any speech recognizer where the misclassification of one or more phonemes will not affect the final decision about recognizing the word. The VA usually used to find the model which yields the maximum value of P (O| λi), and hence, it is used for recognition. In practice, "it is preferable to base recognition on the maximum likelihood state sequence since this generalizes easily to the continuous speech case whereas; the use of the total probability does not. This likelihood is computed using essentially the same algorithm as the forward probability calculation except that the summation is replaced by a maximum operation" (Young et al., 2006).

Figure 2-5 shows the main steps of the VA (Huang et al., 2001), where the $V_t(i)$ is the probability of the most likely state sequence at time t which generates the observation X_1^t (until time t) and ends with state i. the running time complexity of the Viterbi algorithm is $O(N^2 T)$.

As shown in Figure 2-6 this algorithm can be visualized as finding the best path through a matrix where the vertical dimension represents the states of the HMM and the horizontal dimension represents the frames of speech (i.e. time). Each large dot in the picture represents the log probability of observing the frame at corresponding time and each arc between dots corresponds to a log transition probability. The log probability of any path is calculated simply by summing the log transition probabilities and the log output probabilities along that path. The paths are grown from left-to-right column-by-column (Young et al., 2006) (Parhi, 2002).



```
Algorithm Viterbi
{
    Step 1: Initialization
        $V_1(i)=\pi_i b_i(X_i)$,        $1 \le i \le n$
        $B_1(i)=0$
    Step 2: Induction
        $V_t(j)=\max_{1 \le i \le N}\{V_{t-1}(i)a_{ij}\} b_j(X_t)$,      $2 \le t \le T; 1 \le j \le N$
        $B_t(j)=\text{Arg}\max_{1 \le i \le N}\{V_{t-1}(i)a_{ij}\}$,      $2 \le t \le T; 1 \le j \le N$
    Step 3: Termination
        Max Score$=\max_{1 \le i \le N}\{V_t(i)\}$
        $\dot{S}_{T=}$ Arg $\max_{1 \le i \le N}\{B_T(i)\}$
    Step 4: Backtracking
        $\dot{S}_{T=}B_{t+1}(\dot{S}_{t+1})$    t=T-1, T-2,……..,1
        $\dot{S}_{=}(\dot{s}_1, \dot{s}_2, \dot{s}_3, \ldots \ldots, \dot{s}_T)$    is the best sequence
}
```

Figure 2-5: Viterbi Algorithm Main Steps (Huang et al., 2001)

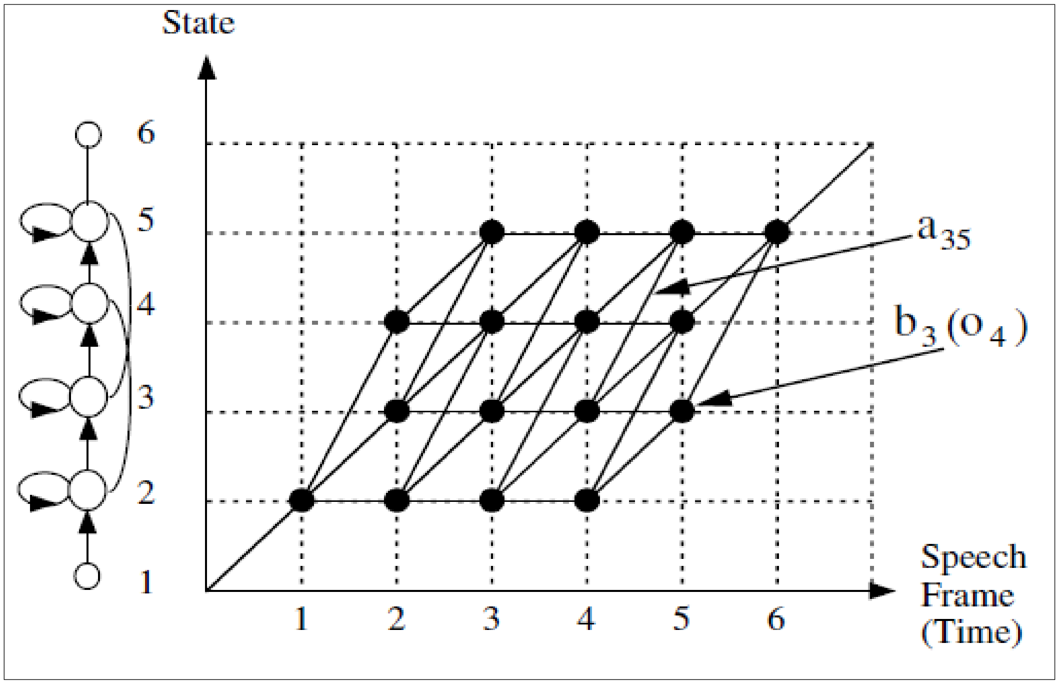


Figure 2-6: Viterbi Algorithm for Isolated Word Recognition (Young et al., 2006)

2.6 K-means Clustering Algorithm

“The term "K-Means" was first used by (MacQueen, 1967), in fact, the idea goes back to (Steinhaus, 1956). The standard algorithm was first proposed by (Lloyd, 1982) as a technique for pulse-code modulation, though it wasn't published until 1982” (Wikipedia, 2013). The algorithm is described mathematically as follows:

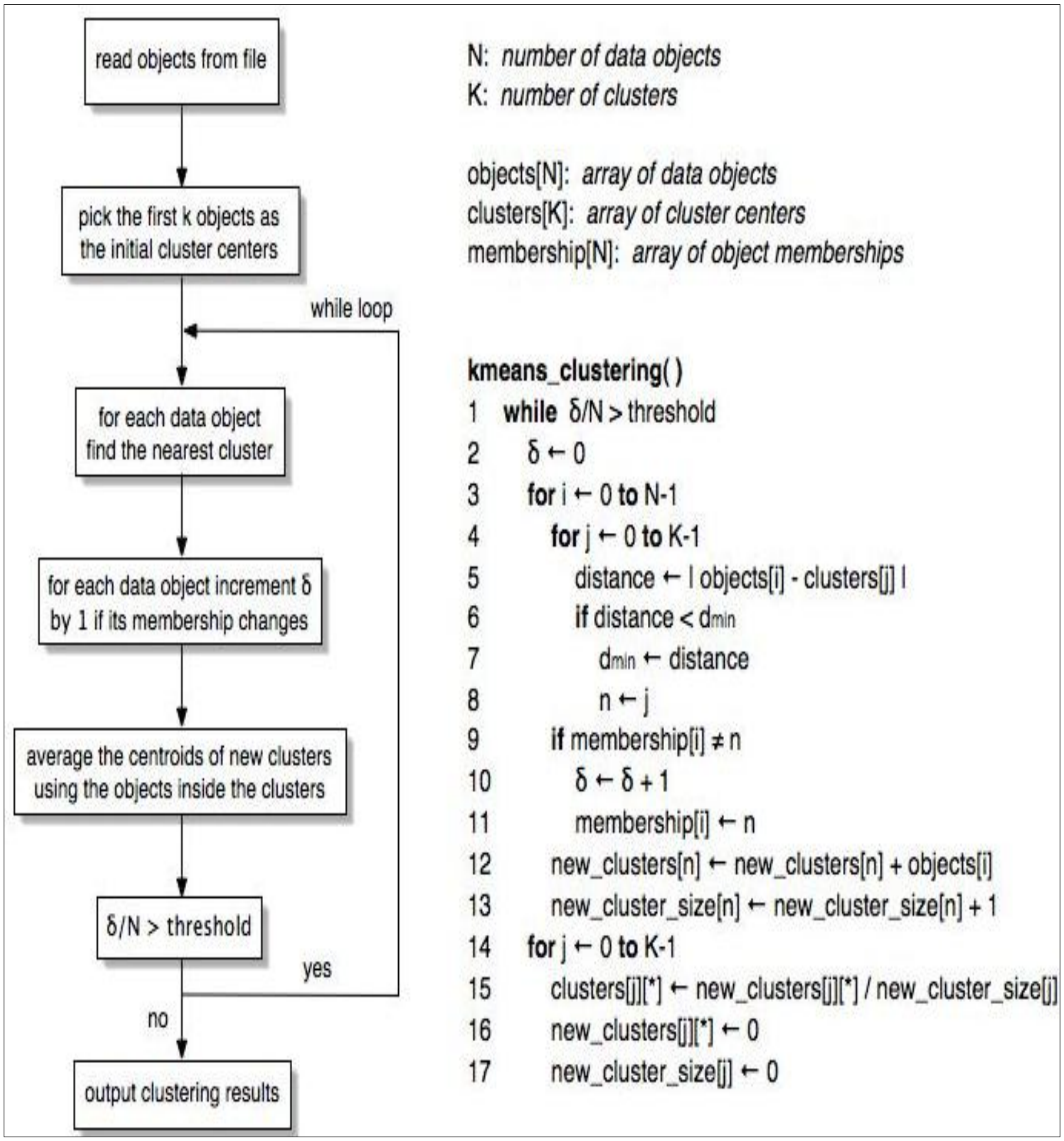
“Given a set of N-observations **$(X_1, X_2, \ldots, X_N)$**, where each observation is a d-dimensional real vector, K-Means clustering aims to partition the N-observations into K sets S, where **$(K \leq N)$ and $S = \{S_1, S_2, \ldots, S_k\}$** so as to minimize within-cluster sum of squares (WCSS)” (Wikipedia, 2013), formally written as:

$$Arg_s Min \sum_{i=1}^{k} \sum_{Xj \epsilon Si} \left\| X_j - \mu_i \right\|^2 \text{ , where } \mu_i \text{ is the mean} \qquad \textbf{(2-16)}$$

The K-Means clustering algorithm is commonly used in computer vision for image segmentation. A weighted distance measure is used in this context for better image clustering. The basic steps of K-Means clustering as described by (MacQueen, 1967) and (Alpaydin, 2004) as follows:

1. Initialize the $\boldsymbol{\mu_i}$ to be the mean of each group (or cluster).
2. Assign each example in the data set to the closest group (represented by $\boldsymbol{\mu_i}$).
3. Recalculate $\boldsymbol{\mu_i}$, based on the examples that are currently assigned to it.
4. Repeat steps 2-3 until $\boldsymbol{\mu_i}$ converges.

(Northwestern University, 2009) draws a complete flowchart along with the algorithm details as shown in Figure 2-7.


read objects from file
pick the first k objects as the initial cluster centers
while loop
for each data object find the nearest cluster
for each data object increment δ by 1 if its membership changes
average the centroids of new clusters using the objects inside the clusters
δ/N > threshold
yes
no
output clustering results

N: *number of data objects*
K: *number of clusters*

objects[N]: *array of data objects*
clusters[K]: *array of cluster centers*
membership[N]: *array of object memberships*

kmeans_clustering( )

```
while $\delta/N >$ threshold
    $\delta \leftarrow 0$
    for i ← 0 to N-1
        for j ← 0 to K-1
            distance ← | objects[i] - clusters[j] |
            if distance < $d_{min}$
                $d_{min}$ ← distance
                n ← j
        if membership[i] ≠ n
            $\delta \leftarrow \delta + 1$
            membership[i] ← n
        new_clusters[n] ← new_clusters[n] + objects[i]
        new_cluster_size[n] ← new_cluster_size[n] + 1
    for j ← 0 to K-1
        clusters[j][*] ← new_clusters[j][*] / new_cluster_size[j]
        new_clusters[j][*] ← 0
        new_cluster_size[j] ← 0
```

Figure 2-7: K-means Algorithm as Described by (MacQueen, 1967) Taken from (Northwestern University, 2009)

2.7 Learning Vector Quantization Algorithm (LVQ)

The LVQ algorithm was developed by (Kohonen, 1988). The LVQ works by dividing the vector space in a pattern classification problem with decision lines that closely optimal as those in Bays Decision rule. It is concerned with perfect classification in attempt to minimize classes' misclassification when class distributions overlapped. The core function of the LVQ, is to find the proper decision lines between classes when they are overlapped or crossed by each other. To separate classes perfectly, the LVQ starts by generating a decision line at the place where the classes' distributions cross.

For the learning to occur in the LVQ algorithm, for a given training vector x, three conditions must be met:

1. The nearest class must be incorrect.
2. The next-nearest class (found by searching the reference vectors in the remaining classes) must be correct.
3. The training vector must fall inside a small, symmetric window defined around the midpoint of the incorrect reference vector and the correct reference vector.

When all three conditions are met, then adaptation movement occur for the incorrect reference vector by moving it further away from the input, and the correct reference vector is moved closer to the input.The adaption equation for m-frame time window and closest reference vector $\mathbf{m_i} \in \mathbf{R^n}$ is:

$$\boldsymbol{m_i(t+1) = m_i(t) + \propto (t)(X(t) - m_i(t))} \qquad \textbf{(2-17)}$$

And the adaption equation for mi $\notin \mathbf{R^n}$

$$\boldsymbol{m_i(t+1)} = \boldsymbol{m_i(t)} - \propto \boldsymbol{(t)(X(t)} - \boldsymbol{m_i(t)}) \qquad \textbf{(2-18)}$$

The symbol $\mathbf{R^n}$ represents the in-class frames. Equation 2-17 means moving closer to in-class by $\boldsymbol{\alpha}$ learning rate while Equation 2-18 means moving closer to Out-class and far away from in-class. Equations 2-17 and 2-18 are usually put in a loop that iterates for a predefined number of iterations or iterates until the desired accuracy is reached. At each iteration the learning rate α is reduced by a specific value depending on the problem. The iterative process is usually called training phase and it is applied to obtain the trained codebooks.

CHAPTER 3

LITERATURE REVIEW

3.1 INTRODUCTION

What is a spoken language? How can human recognize commands, things, names, verbs, ..., etc. when hearing speech in a language? Many questions can be asked on the nature of language and its relationship to recognition by human brain. Language is used when writing and when speaking. It is defined as a system of words and grammars but, the spoken language is a systematic means of communicating by the use of the sounds or conventional symbols (Huang et al., 2001).

Human civilizations on earth used languages for communication between individuals of the same nation and, learned other languages in order to communicate with other nations. In recent civilization where the computer revolution appeared and the big improvement in technology takes over, the machine starts to simulate the human role in almost everything including speech recognition. The speech recognizers appeared, improved and still improving.

Most of the ASR systems depend on a predefined set of phonemes, but rarely depend on set of data driven SWUs. Various attempts were made for automatic determination or extraction of SWUs on the natural spoken languages such as; English language, Arabic language and other spoken languages in the world. No evaluation of the optimality of the knowledge-based Arabic phonemes was performed up to our knowledge.

Our review in this chapter focuses on both the techniques applied on Arabic ASR systems and. The road map of our literature review flows in the following sequence; first, a review on the previous attempts regardless the language type (Section 3.2). Then, we study attempts done on non-Arabic languages and how they identify and solve the problem under study (Section 3.3). Finally, the most relevant Arabic references on our problems were studied and summarized (Section 3.4). A separate section is designated for those papers which are closely related to our research (Section 3.5). Finally, a summary table for the references is given (Section 3.6).

3.2 Early Research on Sub-word Units Extraction

Historically and up to our best knowledge the early attempts done on SWUs started when (Zue and Lamel, 1986) used multiple sources of knowledge in order to determine the phonetic identity. The sources include; articulatory movements, acoustic phonetics, phonotactic, and linguistics. They investigated the feasibility of constructing a knowledge-based system that mimics the process of spectrogram reading by humans. The system achieved good performance comparable to that of the experts.

(Svendsen et al., 1989) devised SWUs for isolated word using maximum likelihood (ML) segmentation. The purpose was to group frames together based on acoustic similarity. A ‘segments quantization’ similar to vector quantization used to cluster ML acoustic segments to N number of clusters. The N clusters were modeled by HMM and a set of units defined and trained using the model. Three methods were devised for lexicon generation; the first one was to choose the pronunciation at random for each word from various utterances and the other two methods were based on clustering.

(Pieraccini and Rosenberg, 1989) built a single model for each phone unit, using clustering. They reported that when increasing the number of models, the accuracy of the system was enhanced but, this enhancement stopped at some point because the desired accuracy threshold was reached. About 450 sentences were uttered by one male speaker as a feed for the estimation process of the Algorithm. Learning was done by assigning weights to different models, in order to account for the context of the unit within words and provide more discrimination among words. Moreover, the training phase consisted of initialization, clustering, model estimation, segmentation and iteration. Svendsen et al. model and Pieraccini and Rosenberg model were not far away from each other but Svendsen et al. model was an improved version for Pieraccini and Rosenberg model with a similar but simpler form of mathematical formulation of the problem.

(Lee et al., 1989) compared three types of fundamental SWUs: whole word units, phoneme-like units and acoustic segment units. They pointed out the advantages of each model. Their research showed that the whole words ensure the integrity of the word and it achieved the best performance among the three types. The recommendation was to use this model whenever short word training data is under consideration. For phoneme like units, the base form dictionary was the best way to build the word model that has not appeared in the training data. For acoustic unit segmentation the consistency of segmentation was preserved by unit modeling and acoustic lexicon. The used dataset was based mainly on TIMIT database, PLU's, DLU's along with 1109 vocabulary English word. Final recommendation of (Lee et al., 1989) was to use hybrid whole word and sub-word model to achieve best results.

(Hatazaki et al., 1989) devised a method for phoneme segmentation using expert system. The expert system used a spectrogram to determine boundaries of the phonemes as well as their categories. The system performed assumption-based inference with certainty factors, and top-down acoustic feature extraction under phonetic context hypotheses. The system was applied on Japanese consonants and was able to detect about 90% of the phonemes correctly. It was as accurate as those detected by human experts. The final result achieved was that the phonemes obtained by the expert system can be identified using a stochastic phoneme recognition method.

(Paliwal, 1990) extended the lexicon generation to be in a probabilistic form. A statistical model of pronunciation was trained on all pronunciation variations of the word in training data. The state of the HMM model represented the SWUs that enhanced the deterministic model.

(A. Ljolje and M. D. Riley 1991) investigated an automatic approach for segmentation of labeled speech. They also, investigated the speech labeling based on the availability of speech transcription. The technique that they used was based on a phone recognition system with different models like: trigram model, gamma duration model and a spectral model of five different structures. Utterance alignment was constrained by the available transcription. Moreover, a prediction tree was used for phone recognition when it was misclassified and the maximum likelihood phone sequence is considered the true one and compared with the reference.

(Bahl et al., 1993) introduced frame-sized units labeled by a vector quantizer (VQ) called ‘fenones’ for recognition of isolated word. 200 fenones were generated by VQ for 5 minute speech. Then, the VQ vector was extracted for each single word at regular intervals and compared to the 200 fenones. Each fenone was given a label based on the Euclidean Distance. The label sequence was the ‘fenonic baseform’ of the word. Fenones were modeled by Markov models and trained using 9 utterances of the same word. This process improved word error rates on small vocabulary of isolated words but, not for large vocabulary. They also modeled pronunciation variations with more flexibility by using more Markov states. Since each word was treated independently to determine the pronunciation, these methods did not require a lexicon generation step. This is not possible unless the word boundaries are known.

(Riley and Ljolje, 1996) explored ways of “spelling” a word in a speech recognizer's lexicon and how to obtain those spellings. They compared three sources of SWUs; coarse, fine and multiple phonetic realizations. They described how to get these different pronunciations using text-to-speech systems. Procedures were trained on phonetically labeled corpora. The methods were evaluated by applying them to speech recognition using DARPA resource management (RM) and North American News (NAB). For RM, they obtained 96.3% word accuracy using multiple phonetic pronunciations per word with associated likelihoods. For NAB, they obtained 90.0% of word accuracy.

(Fukada et al., 1996) incorporated phonetic knowledge to deal with unseen words by devising phoneme models which consisted of a set of SWUs. The devised phoneme models where reached by aligning the SWUs with phoneme transcription. The phone was the merged SWUs that represented phoneme in the data. A dictionary of words was created by concatenation the phoneme models that made up the word's pronunciation. The acoustic segments were automatically defined. The lexicon jointly used these phoneme transcriptions that enhanced the HMM phoneme-based system. It is obvious that the system was still based on the phonetic knowledge.

(Petek et al., 1996) compared three different segmentation algorithms which were all based on variants of the Spectral Variation Function. Ontario Genomics Institute (OGI) produced a multi-language telephone speech corpus. It was used to compare applying improved algorithm in the task of automatic segmentation for spontaneous telephone quality speech to those resulting from superimposing white noise.

(Sharma and Mammone, 1996) used a blind speech segmentation approach. The technique allowed a speech sample to be segmented into SWUs without linguistic information. This approach found the optimal number of sub-word segments in a given speech sample, before locating sub-word segment boundaries. They compared between speech segmentation error rate when the text transcription is available and the blind segmentation, they reached 1.37% of error rate in the first method and 0.93% of error rate in the second method recpectively.

(Westendorf and Jelitto, 1996) presented an algorithm for automatically learning pronunciation variations from speech data. They showed an approach for automatically generating suitable pronunciation dictionaries from the speech database itself. The only knowledge resource that was used in addition to speech database is the unlabeled signals and their transliterations on word level.

(J Cernocky 1998) aimed to determine speech units using Automatic Language-Independent Speech Processing (ALISP). Cernocky's methodology flowed in two steps; first, initial speech unit set was defined and transcription database was obtained where the temporal decomposition (TD), unsupervised clustering and multigrams have been used. Second, HMM was used to model the units with a refinement process to develop the model set. The researcher tested his system on two applications; the first one was with a very low bit-rate speech coding, intelligible speech was obtained at mean 120 bps for unit encoding in two sets of speaker-dependent experiments. The second application was to use those units as pre-processing for segmental speaker verification system. The segmental system showed a comparable performance using those units.

(Kannan and Ostendorf, 1998) compared parametric and nonparametric constrained-mean trajectory segment models for large vocabulary speech recognition. They found that parametric model had fewer free parameters and gave similar recognition performance to nonparametric model, but it had higher recognition costs.

(Kessens, Wester, and Strik 1999) improved the Dutch speech recognizer by adding the pronunciation variants to the lexicon, retrained phone models and using language models to which the pronunciation variants have been added. Within-word pronunciation variants were generated by applying some phonological rules to the words in the baseline lexicon. Two approaches were used; in the first approach, cross-word processes were modeled by adding the cross-word variants to the lexicon. In the second approach, cross-word processes were modeled using multi-words. They tested the combination of within-word method with the two cross-word methods. The combination showed the best results. Word error rate (WER) reduction was about 1.12% compared to the baseline.

(Bacchiani, 1999) recognition system was based on automatically derived units that use ML acoustic segmentation. The pronunciation constraints enable the dictionary to be written directly. Each token had the same unit sequence that made up the pronunciation. Pronunciation was made up of SWUs. It was a process of clustering segments within tokens of a word and across words in advance. Word boundaries were still required for training and the method needed no initial lexicon or phone transcription.

(Furuichi et al., 2000) depend on detecting the boundaries of the phonemes, then discriminating them using stochastic phoneme segment model. In the recognition process matching between symbol sequences to dictionary items was done, which eliminated unnecessary parameters and was based totally on feature parameters, which led to effective discrimination between phonemes. Their proposed model was trained on a small number of training data consisting of 4920 words uttered by 10 speakers. The recognition rate was 92.6% at the word level as the average of all speakers using a dictionary of 212 words.

3.3 Recent Researches on Sub-Word Units Extraction

(Singh et al., 2002) proposed a methodology for extracting English SWUs. The details of this paper will be discussed in Section 3.4 (notable work).

(Sekhar et al., 2003), studied multiple different classification methods for SWUs recognition. They used both large Japanese language corpus for monophone units and

Indian language TV-News corpus for consonants-vowel recognition. They compared Support Vector Machine learning algorithm (SVM) to other algorithms and they achieved a word accuracy of 62% using SVM and 58.4% using other techniques. To avoid loss of information when extracting fixed length patterns due to varying duration segment, they used dynamic alignment kernels.

(Raju, 2003) tackled the problem of language identification in his Master thesis by investigating three main frameworks in phone recognition approaches. She found that these approaches suffer from the limitation of requiring phonetically labeled data for the training of the front-end phone recognizer. She relieved this problem using a sub-word recognizer (SWR) in place of phone recognizer. He obtained the SWR from training data without any phoneme transcription in any of the languages used. He used the SWR as the front end recognizer which was based on the SWUs extracted using automatic segmentation of the training data (segment clustering and HMM segment modeling). Raju studied three sub-word systems for identifying different languages e.g.; English, German, Hindi, Japanese, mandarin and Spanish The researcher concluded that the sub-word language identification systems can operate as well as phoneme based identification systems. Different rates of performance were reported for each language.

(Jiang and Huang, 2003) in their USA patent converted the analog speech signal into digital signal and extracted at least one feature from the digital signal. A word string which consisted of SWUs was identified from the extracted feature. Each extracted word had its own confidence measure scores for each sub-word unit. Different confidence measures were created for different words by applying different weights to confidence scores associated with different sub-words of the identified word string. The weights of the weighted confidence measure scores were determined using the training data.

(Mitchell et al., 2003) overcome the SWUs problem by early determination of onset of a valid spoken input by examining SWUs in a decoding tree. Their invention operates at the sub-word level which made their method more efficient compared to long duration words. There invention also efficiently utilizes CPU processing time since it targets only the best scoring paths instead of checking all possible alternatives. The invention allows for rapid stopping of the prompt to improve automatic speech recognition (ASR) which reduces speaker confusion and or frustration.

(Adami and Hermansky, 2003) proposed a methodology to segment and label the speech signal into small set of discrete units that can be used to characterize speaker and/or language. The methodology was evaluated using National Institute Standard Technology (NIST) extended data speaker detection task and the NIST language identification task. The proposed segmentation made 19% relative improvement over other work.

(Kamakshi Prasad, Nagarajan, and H. a Murthy 2004) presented a new algorithm to automatically segment continuous speech signals in syllable-like segments. They used the group delay processing of the magnitude spectrum to determine segment boundaries in speech signal. TIMIT and TIDIGITS databases were used as a corpus for the experiments. Performance was tested on both continuous speech utterances and connected digit sequences. The segment boundary error was less than or equal to 20% of syllable duration for 70% of the syllables.

(Sreenivas, 2004) proposed a K-means segmental training algorithm (SKM) for design of the Ergodic HMM directly from the language utterances. The algorithm optimized the front-end of sub-word recognizer (SWR) and the back-end language model (LM). Thereby, it was an improved modeling compared to phone-based recognizers.

(Erdogan et al., 2005) used the longest possible SWUs in their lexicon. They incorporate the linguistic rules of the half-words into the Language Model (LM). To yield better and smaller LM, they presented the language constraints with a rule-based weighted finite state machine (WFSM) which can be combined with an N-gram LM. They studied the performance using Turkish large vocabulary recognition system (LVCSR). They found that using half-words with a bi-gram model would significantly reduce the word error-rate as compared to full-word bi-gram model. They also found that, when combining a tri-gram half-word LM with vowel-harmony WFSM the accuracy improved. Moreover, when rescoring the bi-gram lattices. New error metrics were conducted in this research like Half-word-error rate (HWER) and stem-error-rate (STER). In this research, the authors had introduced novel methods for determining sub-word lexicons and developed LM's for Turkish LVCSR.

(Sadohara et al., 2006) proposed a method for merging similar adjacent parts of sub-word sequence obtained from continuous sub-word recognizer. The aim was to produce a hierarchical cluster tree. A string kernel was used to compute the similarity between two sub-words units. They excluded the influence of the sub-strings that were irrelevant to the topic of interest and topically coherent clusters formed without linguistic knowledge. They carried an empirical study on a Japanese news speech corpus which shows better performance compared to topic segmenter using LVCSR. This study produces a domain-independent topic segmentation algorithm for spoken documents that segments recognized sequences of SWUs without using any LVCSR system.

(Walliczek et al., 2006) used SWUs as prerequisites for LVCSR which allowed the recognition of words not seen in the training set based on seen SWUs. They reported their experiments with syllables and phonemes as SWUs. They also developed a new feature extraction method which improved words and SWUs. They based their tests on electromyography (EMG) speech recognition system where the only input was the electric signals caused by the articulatory muscles which means silent speech (no acoustic signal is produced). They used the 45 English phonemes and 9000 syllables in the training. The tests showed that it is possible to build an EMG speech recognizer based on SWUs, and it is possible to recognize unseen words using SWUs by 62.4% rate.

(Hirsimäki et al., 2006) presented a language-independent algorithm capable of discovering words fragments in unsupervised manner from text. The algorithm uses the Minimum Description Length (MDL) to find word fragments that models the training text effectively. From their experiments done on the extracted inventory, they showed that the n-gram recognition done based on this inventory is better than n-gram based on words. Two Finnish language speech recognition experiments were done based on word fragments found by the algorithm and cause an error rate reduction of 12% and 31% respectively.

(Zhang and Nakamura, 2006) extracted an efficient phoneme set of tone-dependent SWUs by merging a pair of tone-dependent units according to the principle of minimal loss of the mutual information (MI). The MI as it was defined by this research was measured between the word tokens and their phoneme transcriptions in a training text corpus. They selected the phoneme set by balancing between MI and number of phonemes which was very flexible. Different phoneme sets were derived with different number of units. Moreover, different speech recognition experiments were carried out on this phoneme sets and the derived sets showed their effectiveness. The tone-dependent phoneme set was used for building Chinese LVCSR system using MI based criterion.

(Rotovnik et al., 2007) created LVCSR for the Slovenian language that belongs to the group of inflectional languages with rich morphology. Such a language could cause a problem for the LVCSR compared to uninflectional languages like English language. Slovenian language requires a vocabulary approximately 10-times greater for same degree of text coverage. To overcome the out-of-vocabulary (OOV) words problem which is common in the inflectional languages and it has a bad impact on the recognition system, a new searching algorithm was developed by restricting the correct order of the SWUs. The search algorithm combined the properties of sub-word based models (reduced OOV) and word-based models (length of context), it increased recognizer accuracy with comparable search space to standard word-based recognizer. The methods were evaluated using SNBI speech database.

(Pan et al., 2007) improved the Position Specific Posterior Lattices (PSPL) compact structure which was used to index speech to sub-word based position specific posterior lattices (S-PSPL). They included posterior probabilities and proximity information based on SWUs rather than words. Rare and OOV were handled in S-PSPL but not handled in PSPL. They carried out some experiments on Mandarin Chinese broadcast news, S-PSPL showed significant improvement compared to PSPL.

(Tejedor et al., 2008) proposed the direct use of the graphemes (i.e., letter-based SWUs) for acoustic modeling. They used this method on the languages where letter-to-sound mapping being very regular but the correspondence is not one-to-one. They compared three approaches for Spanish keyword spotting or spoken term detection: hybrid approach, 1-best approach and lattice-based approach In each approach they compared acoustic modeling based on phone and grapheme units. In their experiments they used the Spanish geographical-domain Albayzin corpus. The results showed that grapheme-based units for acoustic modeling match or exceed the performance of the phone-based acoustic models. In the experiment of keyword spotting the results were similar in all.

(Kenney, 2008) investigated the automatic determination of SWUs for automatic speech recognition. The method is based on Joint Multigrams (JMs); and included “co-segmentations of sequences of data from two data streams”. The probability of the co-segment was calculated by how often they occurred in data. The final product of such method was a probabilistic dictionary after applying this methodology on streams of words and SWUs. This method did not rely on any linguistic knowledge and no initial dictionary was required. The process used in Kenney's dissertation work to produce SWUs followed the following steps:

1. Connecting N HMM states as an ergodic HMM.
2. Initializing the mean and variance of each state so that they were near the mean and variance of the data but distinct from other states.
3. Training the ergodic HMM on all training data. Did not provide transcription information neither on word level nor on sub-word level.

4. The sub-word unit transcription was the most likely state sequence of ergodic HMM for each utterance. Standard decoding techniques was used to find the most likely state sequence.
5. SWUs were extracted from ergodic HMM, where each state represents one sub-word unit.

With pronunciation dictionary in terms of these models a standard recognition task could be carried out.

Every step of the previous steps has its own details and tests.

(Aksungurlu, 2008) in his thesis work carried out large number of LVCSR experiments mainly based on SWUs like morphemes and stem-ending on Turkish language. These experiments were to overcome the inadequacy of the traditional word based language models that give satisfactory results for English language but are not adequate for the Turkish language. He derived a set of SWUs in unsupervised manner. Then, he investigated their use. Different segmentation approaches were used and compared based on best WER obtained from the recognition. The results achieved, a WER of 25.24% using SWUs compared to 26.90 % using word-based languages.

(Szoke et al., 2008) compared between sub-word based methods (which is used to search for OOV) for spoken term detection (STD) and phone recognition. They compared between words, phones and multigrams and they carried their evaluation on NIST STD06 dev-set CTS data. Two constrained methods of multigrams training were proposed. The results showed that the proposed method improved the phone accuracy more than 9% and STD accuracy more than 7%.

(Majewski, 2008) proposed a syllable-based language model which is more appropriate for highly inflectional language like polish. He found that syllable-based model outperforms word-based models in terms of small corpora and OOV words. He evaluated and showed the usefulness of the syllable based model in speech recognition.

(Ariya Rastrow et.al. 2009) proposed a hybrid system combining words and data-driven variable length SWUs for detecting regions with OOV words in the output of LVCSR. They have reported that the single feature method with its posterior probability that they used outperformed existing published methods. They showed that addition of other features such as word and sub-word entropy helps in improving false alarm regions.

(Sarada et al., 2009) , segmented and labeled continuous Indian speech into syllable-like units. The segmentation was carried out using delay based algorithm. Similar syllable segments were grouped together using unsupervised and incremental training (UTT) technique. They generated isolated HMM models for each cluster during training. During testing the speech signal was segmented into syllable-like units which are tested against the HMMs obtained during training. The results were a syllable recognition performance of 42.6% and 39.94% for Tamil and Telugu respectively. They also proposed a new feature extraction technique that uses features extracted from multiple frame sizes and frame rates during both training and testing. The performances with the new feature extraction technique were 48.7% and 45.36% respectively.

(Choueiter, 2009) focused on modeling and learning lexical properties of OOV words. Using context free grammar, she built a linguistic model that describes sub-syllabic structure of English Words supported by a probability model. The probability model captures the statistics of the parses generated by the grammar and encodes spatial-temporal context. From the context free grammar, she extracted the SWUs, which encode pronunciation information. She introduced a hybrid entity called spellneme-unit which was used in the design of the bi-directional letter-to-sound (L2S) model. L2S was used in automatically learning the spelling and pronunciation of a new word. Both SWUs and L2S model were assessed on the task of automatic lexicon generation. He used SWUs in a flat hybrid OOV model for continuous ASR. The performance of the hybrid front-end recognition system reported in terms of word and sub-word error rates and it outperform word-only system. Moreover, the hybrid system outperforms alternative linguistically-motivated sub-lexical units such as phonemes.

(Thangarajan et al., 2009) provided a solution for Tamil language. The Tamil language suffers from challenging features like agglutination and morph-phonology. They used syllable as sub-word unit in the acoustic model. They developed an algorithm based on prosodic syllable and carried out two experiments based on it. In the first experiment, syllable based context independent models had been trained and tested. It performed reasonably well compared to context independent word models in terms of WER and OOV words. In the second experiment, integration was carried out between the syllable information in conventional triphones modeling were cross-syllable triphones replaced with monophones and the number of context dependent phone was reduced by 22.76% in united units. Analysis of 2.6 million word corpus showed that only 4023 syllable occurs more than 25 times. The number of models to be trained was reduced and a 10% reduction of WER was achieved and OOV word rate was also reduced.

(Rastrow et al., 2009) augmented a word-based system with SWUs as a step towards open vocabulary speech recognition systems. Their system combined words and data-driven, variable length SWUs. They have reported that data-driven SWUs have better phone accuracy than words only systems and data-driven SWUs are better in detecting OOV terms. They used RT-04 broadcast news and MIT lecture data sets to carry out their experiments.

(Zhang, 2009) presented two unsupervised spoken keyword spotting systems without the need for any transcripted data. The first system trained the Gaussian Mixture Model to label speech frames with Gaussian posteriograms. Segmental dynamic time wrapping was used in this system to compare between keyword samples and test utterances. The second system used the Joint- Multigrams model to avoid the need for spoken samples. The author used the Joint-Multigrams to build a mapping from the keyword text samples to the Gaussian components indices. He measured the similarity score of the Gaussian component index sequences between keyword samples and test utterances. The previous systems were evaluated on TIMIT and MIT lecture corpus. He also presented some preliminary investigation on the unsupervised detection of acoustically meaningful units in speech.

(Pellegrini and Lamel, 2009) enhanced the Morfessor algorithm by a data-driven word decomposition algorithm that incorporates phonetic properties and some constraints on recognition units derived from forced alignments experiments. The aim of the enhancement was to address the problem of increased phonetic confusability arising from word decompounding. They carried out some experiments on the Amharic language speech recognition for broadcast news in order to validate the approach. The OOV word rate was reduced from 35% to 50% with a small reduction in WER. The algorithm was language independent and requires some little adaptation to be applied on other languages.

(G. Aimetti1, Roger K. Moore et.al 2010) presented a computational model to learn words. The model was made up from emergent SWUs with no prior linguistic knowledge. They used the acoustic DP-n-gram algorithm for segmenting the raw acoustic speech signal. The word models were extracted as a sequence of segmented units. They proved that the system can automatically build reusable acoustic SWUs with no predefined language specific rules. They recorded the speech of three native English speakers (2 males and one female), they uttered 120 examples of each of the pair words "Stalagmite" and "Stalactite" recorded at frequency of 16 KHZ and manually segmented.

(Iwami et al., 2010) proposed sub-word unit based recognition and retrieval system. They applied their method on Japanese spoken document retrieval which considered OOV words and misrecognized SWUs. They used individual-syllable as sub-word unit in continuous speech recognition and n-gram sequence of syllables as a retrieval unit. They applied this method to academic lecture presentation database of 44 hours, 60% of the OOV words were detected in less than 2.5ms.

(Barroso et al., 2011) focused on the selection of appropriate SWUs when under-resourced noisy conditions environment exists. The work done was on Basque Broadcast news where they applied several data optimization methodologies to reduce the impact of lack of resources. Hybrid approaches of Discriminate and principle components analysis, robust covariance matrix estimation methods, SVM, HMM, and cross-lingual strategies were used in selecting the SWUs. The accuracy rate was improved for small sample sets and the new methodology provided an excellent tool to manage under-resourced languages, as the authors reported. The best results were obtained by the SVM.

(Baghdasaryan and Beex, 2011) presented a speaker independent continuous speech phoneme recognition and segmentation system. They investigated the feature extraction, constructing and initializing the phonemes HMM models, and automatic segmentation of speech into phonemes. They used different algorithms for training and segmentation processes like: Viterbi algorithm for best lattices searching, Baum Welch algorithm for training and initializing HMM models, K-mean algorithm for segmentation and clustering, and expectation maximization (EM) algorithm for initializing the Baum Welch algorithm. Based on the experiments done on TIMIT speech database the best recognition rate recorded for isolated phonemes was 54.5% For phoneme sequences in continuous speech signal the best recognition rate recorded was 35.41%.

(Duran et al., 2011) developed a framework for solving the correspondence problem (different utterance manifestations).They introduced Correspondence-by-segmentation Hypothesis. The correspondence is achieved by: first, unsupervised segmentation of speech and then mapping the acoustics of speakers onto each other. They demonstrated their hypothesis and showed that different instances of a word can be mapped onto each other with high accuracy when trained on utterance-label pairs for a small set of words.

(Venkatesh et al., 2011) presented a limited vocabulary isolated-word recognition system based on HMM with two stage classification. They proposed a simple metric for segmenting the words into SWUs that were used in the second stage (stage 2) to improve recognition accuracy. Moreover, they presented a simple efficient way to handle OOV words by adding more HMM models. They achieved a recognition accuracy of 89% at the word level using Texas instrument (TI) DA Vinci platform which is an embedded platform for home infotainment system.

(Parada, 2011) presented a methodology for transcribing OOV words when they are identified during the recognition process. The result of this methodology was a hybrid word/sub-word system which predicts full-words for invocabulary OOVs using sub-lexical units for OOVs. The sub-lexical units were automatically learned from data which were variable-length phone sequences included in the vocabulary and language model. She proposed a novel unsupervised approach to learn the sub-word lexicon optimized for a given task. In summary, Parada’s dissertation considers the OOV problem as a sequence labeling problem that could be solved in three main steps:

1. Jointly predicting out-of-vocabulary regions.
2. Including contextual information from each region.
3. Learning sub-lexical units optimized for this task, which leaded to substantial improvements on state-of-the-art systems.

(Xie et al., 2011) used the SWUs in Chinese broadcast story segmentation. They studied the effectiveness of the SWUs in story segmentation of Chinese speech recognition transcripts. They focused on sub-word based Text Tiling and lexical chaining approach to story segmentation where the lexical cohesion was measured, either character or syllable n-gram (n=1, 2, 3, 4). They found that sub-word unigram and bigram outperform word based methods. The testing was done on CCTV corpus and the results showed that sub-word methods often gave better segmentation than word-based on both error free and erroneous transcripts.

(Huijbregts M et al 2011) presented a method for automatic generation of the SWUs which was used instead of phone models in query-by-example spoken term detection system. When given a speech recording a set of speaker models was generated using diarization system unsupervisedly without the need for training or data development. The speaker dependent SWUs were detected based on minimum segment duration. The authors proved that their proposed system outperforms the phone-based system on both broadcast news and non broadcast news by mean averages of 28% and 38% respectively.

(Monica and Nagarajan, 2011) proposed an algorithm to determine phoneme boundary in a given speech signal without the need for transcription. The algorithm was divided into two stages where in stage one, the boundaries were determined by silence/voiced/unvoiced classification. In the second stage the voiced part was tokenized further. TIMIT database was used to carry out the experiments and to check correctness. The maximum accuracy achieved for identifying correct phoneme boundaries was 75%.

(Rahman et al., 2012) used a blind speech segmentation procedure to segment continuous Bangla speech into words/sub-words like units using end-point detection technique. They classified the segmented words based on the number of syllables and the size of the segmented word. They used a windowing technique to extract features. They achieved 96% of segmentation accuracy.

3.4 Recent Research on Arabic Sub-Word Units Extraction

(Awais et al., 2004) built a multilayer recognition system for Arabic phonemes. The first layer focused on classification of the spectra using self organizing maps (SOM), for identifying each Arabic phoneme. In case of multiple classifications of same input sound file, more than SOM were generated for the same input sound file. The second level recognition was based on identifying the pitch cluster to which the input sounds belong. The accuracy when using the first level alone was 71% and 91% when using both layers. The research has focused on recognition rather than determination of the phoneme and was based on a knowledge-base of the phone set.

(Tolba et al., 2005) proposed an algorithm for Arabic speech consonant and vowel (C/V) segmentation. The proposed algorithm did not need any linguistic information; it depended totally on the wavelet transformation and spectral analysis at C/V transition. The authors applied the algorithm on 20 Arabic words recorded six times. They achieved 88.3% for C/V segmentation.

(Awadalla et al., 2005) tried to set some rules and specifications for Arabic database collection to be used in Arabic speech recognition. They studied the Arabic language morphologically and linguistically in attempt to deeply know the properties of each Arabic phoneme, for use in building Arabic speech database. In their work, the Arabic language phonemes were presented and a manual Arabic speech segmentation and transcription were explained

(Anwar et al., 2006) developed basic speech segmentation algorithm for Arabic speech recognition. They dealed with the problem through a number of possible cues, that could help in identifying the phoneme boundaries. They applied some experiments on different number of cues, the best results for phoneme level segmentation were through the power spectral density algorithm (PSD) and Zero crossing rate algorithm (ZCR). The pitch and intensity helped in determining the pauses. They achieved 89% accuracy on the phoneme level for eight different speakers.

(Awais et al., 2006) developed a phoneme segmentation algorithm based on Fast Fourier Transformation (FFT) spectrogram. The output of the algorithm was the phonemes (identified as pauses, vowels and consonants) boundaries. The test bed used was 10 male Arabic speakers voice. They achieved an overall segmentation accuracy of 95.39%. Figure 3-1, reproduced from (Awais et al., 2006), shows the overall implementation steps of the system.

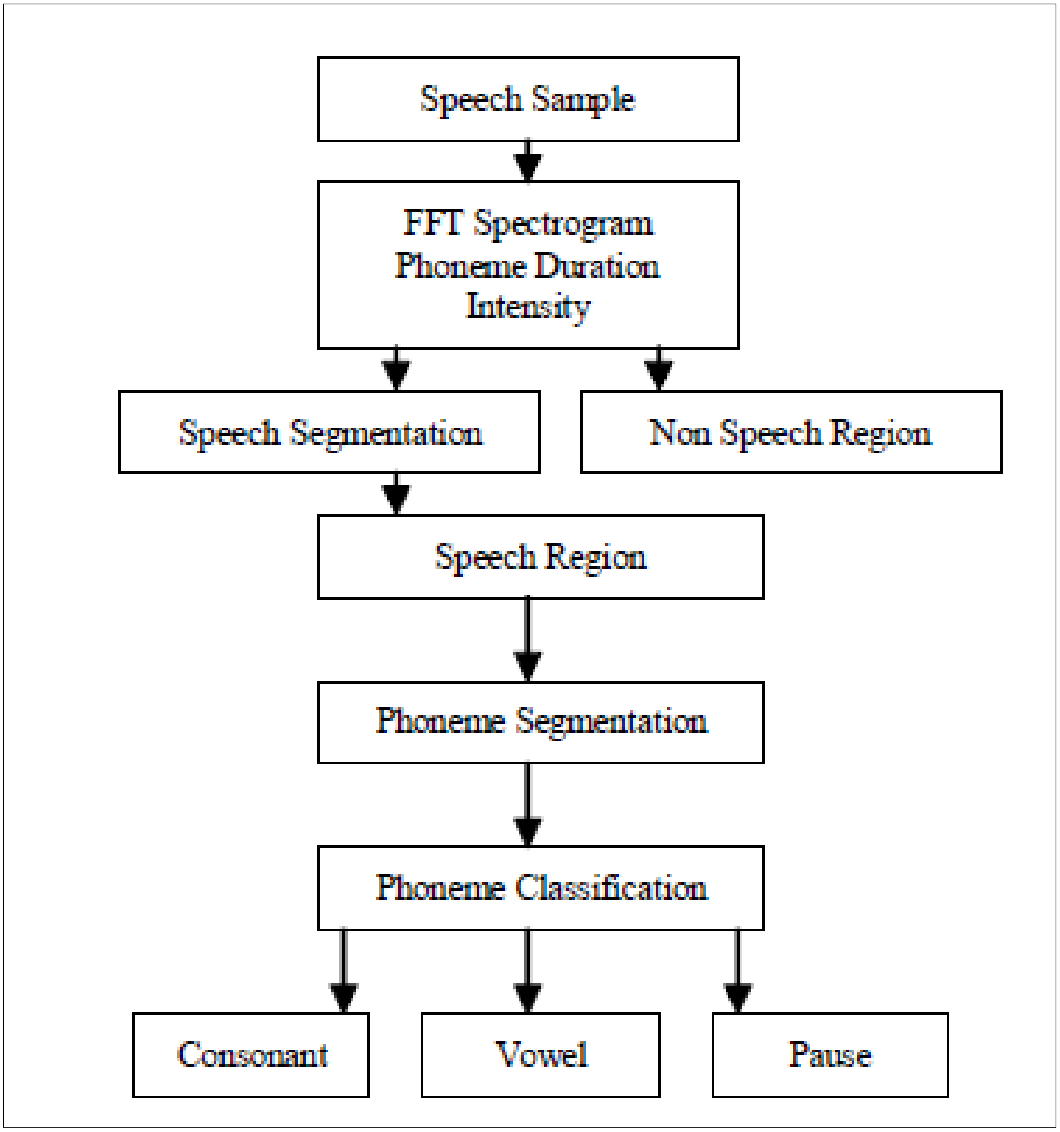


Figure 3-1: Abstract Level Architecture of Speech Segmentation System (Awais et al., 2006)

(Alghamdi et al., 2007) proposed a labeling schema to cover all Arabic sounds including the Quranic sounds. To cover all the phonological variations of Quranic sounds, a set of labels was used and the phones used were the KACST phoneme set database. The dependency was on a predefined phone set.

(Creutz et al., 2007) investigated the morph-based language modeling for LVCSR using four morphologically rich languages: Finnish, Estonian, Turkish, and Egyptian Colloquial Arabic. They applied an unsupervised data-driven method to extract the SWUs (morphs). They used the Morfessor algorithm which was based on estimating n-gram models over sequences of morphs instead of words. Since the standard word models suffer from high OOV rates, the morph models had overcome this problem by concatenating consecutive extracted morphs with low error rates. The Arabic language was the only language that the standard word models outperform the morph models.

(Ali, Elshafei, Alghamdi, et al., 2008) presented a technique called rule-based technique for generating Arabic dictionaries which was used in large vocabulary speech recognition system. The set of phonemes used in building the dictionary were chosen based on the previous experience with Arabic text-to-Speech systems and the corresponding phoneme set which was successfully used in the English ASR. The Arabic dictionary was totally built using this set of phones that were selected by previous human experience.

(Iqbal et al., 2008) provided an algorithm for vowel identification that used formant frequencies. The algorithm recognized the vowels based on extracted formants of already segmented recitation audio files. The investigation was done based on Holy

Quran Tajweed rules. The main objective was to identify Fat'ha (zabar /a/), Kasra (zair /e/) and Damma (pesh /u/) mistakes of the recitor during recitation. Corpus recitation of five experts was used to validate the results. The system showed average accuracy of 90% on continuous speech files of 1000 vowels.

(Azmi and Tolba, 2008) studied the continuous Arabic speech recognition in different noisy conditions. Speaker-independent HMMs were built using the HTK tool kit. The models were trained based on fifty nine Egyptian speakers where each one was asked to utter different sentences of Egyptian proverbs. Experiments were pursued using monophones, triphones and syllables. The results showed that syllables outperform monophones and triphones by 21.46% and 15.63% respectively when signal to noise ratio (SNR) is 20 dB. The authors reported that the result was a big indicator for the usefulness and robustness of syllables based recognition in noisy conditions.

(Sofya and Al-Obadi, 2008) produced a segmentation algorithm of Arabic speech signal into syllabic units. The algorithm was based on the energy and statistical acoustic parameters with the Arabic language phonology characteristics. They tested the algorithm based on the data collected from four speakers (2 males and 2 females). They achieved 85% of accuracy when mono-syllabic and poly-syllabic words used to test the segmentation.

(Al-Manie and Alkanhal, 2009) compared the manual speech segmentation for Arabic speech with the automatic method that depends on the energy. They divided the Arabic phonemes into two energy regions. In the manual approach, a specialized phonetician assisted by using some tools like spectrogram. For the energy approach, Arabic phonemes were clustered based on their acoustic manifestations. Human experience is obvious in both methods.

(Damien et al., 2009) proposed a classification method of modern, classical Arabic (MCA) language and established its phoneme-viseme. The study was based on the geometrical movement of the front lips of four speakers pronouncing predefined sequence of words. The results showed that it was possible to recognize a vowel and determine whether it is a short vowel or a long one, regardless of the consonant.

(Biadsy et al., 2009) proposed a system capable of automatically identifying Arabic dialect (Gulf, Iraqi, Levantine, Egyptian and MSA) of a speaker after training it on his/her speech. They built their own phoneme recognizer using HTK-tool kit and based on 3-state HMM without any skipping state. They combined their recognizer with other languages phoneme recognizers that enhance the Arabic phoneme recognizer. Their phonotactic approach achieved an overall accuracy of 81.60% on the word level using 30s test utterances.

(Heintz, 2010) introduced a method to derive morphemes from Arabic words using stem patterns. Small amounts of linguistic knowledge were used. Her proposed approach (stem pattern derivation) provided a solution for OOV word problem. It was tested in the task of ASR and compared to other three morpheme derivation methods as well as word-based language model. Heintz carried out some experiments on MSA and Levantine Conversational Arabic data. She found that knowledge-light methods of morpheme derivation may work as well as knowledge-rich methods. Moreover, she found that morpheme derivation methods that result in a single morpheme hypothesis per word result in stronger models than those that spread the probability mass across several hypotheses. In addition, she exploited an FSM with which the stem pattern derivation method was implemented to predict short vowels and the result was exploring the use of the morphemes in Arabic language modeling for automatic speech recognition.

(Novotney et al., 2011) attempted to adapt MSA model to Levantine Arabic without manual transcription. They showed that self-training was not a suitable solution to improve strong MSA models on Levantine due to model bias.

(Abdul-Kadir and Sudirman, 2011) discussed the correct and simple way of Arabic phonemes pronunciation in Malay accent. The international phonetic alphabet of Arabic chart was used as a reference for Malaysian children recorded samples. The consonants of standard Arabic phonemes were investigated and articulation of every one was measured. They found that only seven out of 25 consonants of standard Arabic phonemes were considered difficult to utter by Malaysian children. The obtained values were used as reference of the database and for the recognition process. (Abdul-Kadir and Sudirman, 2011) referred to the work done by (Awais et al., 2004) to use the rules and specifications of the phonemes.

3.5 Notable Sub-Word Units Extraction Systems.

(Singh et al., 2002) formulated a complete probabilistic model for extracting the sub-words for continuous speech recognition in the English language, based only on the input training data and their transcripts. The authors did not include any prior knowledge in their experimental part, but they allowed outside resources like: word boundaries, small lexicon...Etc, if they exist, to be incorporated in the solution. They assured that, if no outside resources then the results would not be affected. The framework results could be improved if knowledge existed about the word boundary, but this is not critical for the solution of the problem.

Figure 3-2, summarizes the algorithm. The authors provided a mathematical derivation of the problem and an algorithm that form up a good base for a data driven phoneme set. The algorithm starts by initializing a small number of phonemes (n). As the algorithm progresses, phonemes could be incremented so that they maximizes the recognition accuracy or converges towards an accuracy threshold. Regarding the optimal dictionary and acoustic model that have to be supplied at the beginning, the paper suggested using the alphabet to initialize the dictionary or representing the words in the dictionary by repeated single symbol based on the length of the words. Moreover, the phone set size could be increased by clustering data corresponding to each phone (obtained through phone segmentations) from current acoustic model and dictionary. The dictionary and the model are iteratively updated. A fixed dictionary is used to find the best word segmentation. A check on the new phone updates and the accuracy of the recognition system is carried out as the next step to see if it converges to likelihood accuracy or not. In this case either the system stops or continues by increment (n) and finds the optimal dictionary or model updates. The dictionary updates are mainly based on word segmentation updates that will enhance segments based dictionary updates. They used a statistical correlation between the spelling of a word and its pronunciation. The generated sub-words units and dictionary results from this platform were based only on the corpora of 2.7 hours. Their methodology performed worse than manually designed SWUs. The authors justify this result by referring to the human experience used in manual derivation. The methodology suggested that the existence of external resources added to the algorithm like knowing the boundaries of the words for example, could results in more promising recognition accuracy. In fact, they did not support the algorithm by any external data and in case they did so, this might assist the process of phoneme set detection.

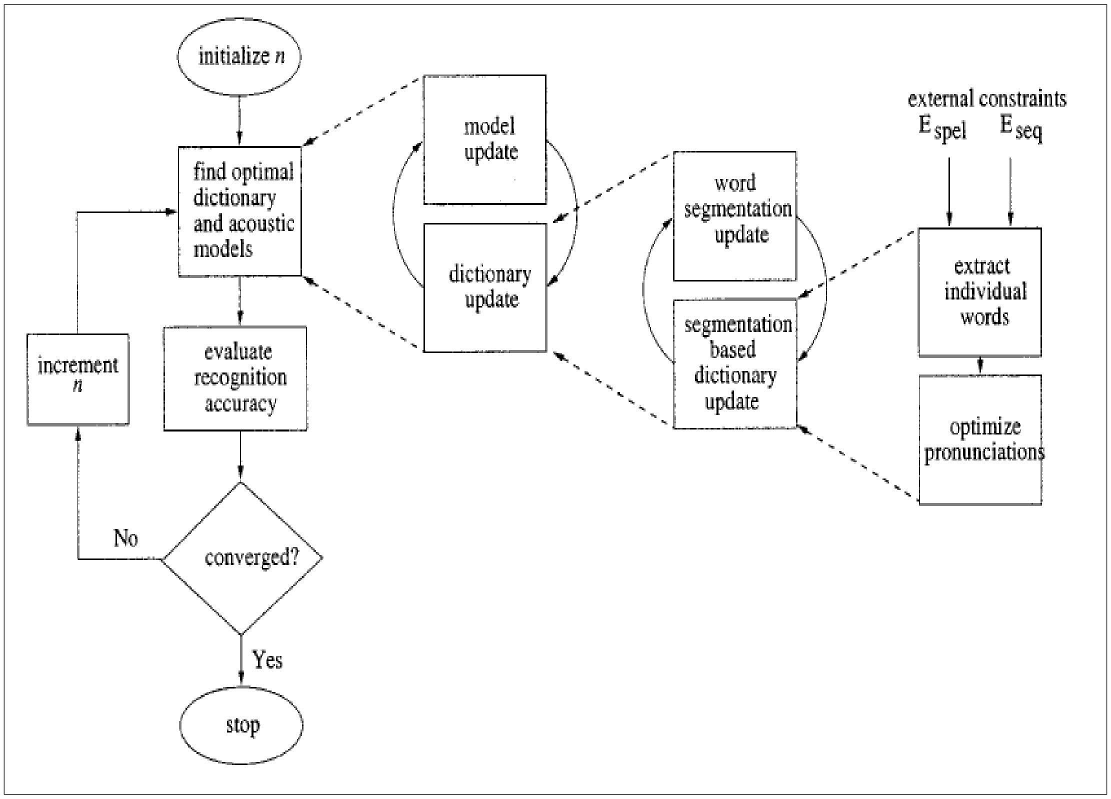


Figure 3-2: Automatic generation of SWUs and dictionary (Singh et al., 2002).

(Singh et al., 2002) is one of the main references in our research for Arabic SWUs extraction. The main steps of the methodology used in this paper are summarized as follows:

1. Initialize the phone set. For example, assume the Phone set is equal {a, b, c, …., z, A, B, C, …, Z};
2. Generate the dictionary by rule based manner using the corresponding alphabet. For example the transcript of the word “CAR” is /C/A/R/.
3. Train the phone models using the dictionary.
4. Generate word segmentations and for each word generate phone codes.
5. Create a pronunciation graph using Baum-Welch algorithm, and choose N-best pronunciations.
6. Generate spelling to phone mapping using Viterbi algorithm for step 4, then identify the most likely pronunciations in this step.
7. Update the dictionary.
8. Check if the Maximum Likelihood of words converged?
9. Go to step 4 if the answer in step 8 is negative.
10. Test recognition accuracy on held out data if it does not reach a predefined threshold.
11. Train the phone models on the newly updated dictionary by going to step 3 again.
12. Update the phone set size, by asking if the accuracy gained from out data and current phone set size is greater than the accuracy when using previous phone set size. If the answer is no we stop because we have reached the best set of phones else we proceed to step 13.

13. Split most frequent phones and go back to step 2 and repeat the process.

Some similarity exists between T.Svendsen models (1989) with the one done by Singh et al (2002), especially in the mathematical formulation of the problem with some variation. The training phase of Svendsen model consisted of a set of steps; “first, segmentation of utterances into acoustic segments. Second, clustering the segments gained from first step into N clusters. Third, labeling of the acoustic segments into sub-word classes and forming sub-word clusters. Fourth, generating of the HMM model for each sub-word from the acoustic segments and finally, the generation of the acoustic lexicon based on maximum likelihood decision rule” (Svendsen et al., 1989). The input utterance is compared with different word models where each one consisted of SWUs. About 100 repetitions of 42 Norwegian speakers formed the dataset; it was fed to this system. This dataset was recorded during 5 weeks on a sample rate of 8 KHZ with low pass filter and cut off frequency of 3.5 KHZ. There was a reasonable reduction in the complexity of the system compared to the baseline system and to Dynamic Time Wrapping (DTW) speech recognizers.

3.6 Summary and Discussion of Literature Review

The literature review started by presenting the initial research attempts for SWUs extraction research in this field. Most of Arabic related work addressed segmentation and linguistic phonemes that are fully depend on human linguistic knowledge in this field. A summary for the most related Arabic researches is given in Table 3-1.

Table 3-1: Summary of Arabic Research on SWUs Extraction.

Author	Purpose	Depends On
Mian M.Awais et al. (2004)	Multi-Layer Phone Recognition System	Knowledge Based
Awadalla et al. (2005)	Deeply Knew Arabic Phoneme Properties And Setting Rules And Specifications for Arabic database collection used in Arabic Speech Recognition System.	Arabic Morphological and Linguistic Analysis
Mohamed Ali et al. (2006)	Rule-based Technique For Generating Arabic Dictionary	Chosen Phones
Mian.M Awais et al. (2006)	FFT in a Phoneme Segmentation algorithm	Database Of Phone Set
Alghamdi et al. (2007)	New Labeling schema for Quranic Sounds	KCAST Phone Set
Iqbal et al. (2008)	Recognizing Vowels Identification That Uses Formant Frequencies	Holy Quran Tajweed Rules.
Mohamed Al-manie et al. (2009)	Comparison Between Manual Speech Segmentation and Automatic One Based On Energy	Database of Phones
Abd Al-kadir et al. (2011)	Correct Arabic phoneme pronunciation in Malay accent	Phonetic Alphabet of Arabic Chart
Abu Zeina et al. (2011)	Cross-word Arabic pronunciation variation modeling	KCAST Phone Set

We reviewed also different techniques in automatic extractions of SWUs; most of these techniques were applied on English language. We will compare between these approaches in attempt to identify the best way to move towards our target in “Automatic Extracting of Arabic SWUs for Continuous Speech Recognition”. In early systems represented by Bahal et al. (1993) and T.Svendsen et al. (1989), recognition was done on isolated word data and the word boundary investigation was completely avoided.

The techniques used in K.K. Paliwal (1990), T. Fukada et al. (1996), M.A. Bacchiani (1999) and Singh et al. (2003) required word boundaries as input . Table 3-2 summaries the methods used and what tasks they focused on and the acoustic models used with their constraints.

Table 3-3 summarizes the recent research done on sub-word units extraction.

Table 3-2: Analysis of Early Research on SWUs.

Author	Purpose	Depends On
Data Driven Approaches		
Sharmma et al. (1996)	Segmenting Speech Samples Into SWUs Without Linguistic information	Data Driven Approach
Westendorf (1996)	Algorithm for Automatically learning Pronunciation Variations from speech	Data Driven Approach
Petek,1996	Comparsion of Three Segmentation Algorithms	Data Driven Approach
(Zue and Lamel, 1986)	Phonetic Identity	Spectrogram
(Svendsen et al., 1989)	SWUs For Isolated Word	Randomness
(Pieraccini and Rosenberg, 1989)	Single Phoneme Using Cluestring	Knowledge-base Approach
(Lee and Hon, 1989)	Comparsion study of SWUs	Knowledge-base Approach
(Hatazaki et al., 1989)	Phoneme Segmentation	Spectrogram for Phone Boundary
Bahal et al. (1993), Paliwal (1990)	Isolated Word	Knowledge-base Approach
Loliji et al. (1991)	Automatic approach for segmentation of labeled speech	Knowledge-base Approach
Riley et al. (1995)	Ways of spelling a word in a speech recognizer and how to obtain those spelling	Knowledge-base Approach
Fukada et al (1996)	Isolated Word	Knowledge-base Approach
(Riley and Ljolje, 1996)	Spelling of Words	Knowledge-base Approach

Petek et al. (1996)	Compared Different Segmentation Algorithms	Knowledge-base Approach
Cernocky (1998)	Determined Speech Unit Using ALISP	Knowledge-base Approach
Pellom et al. (1998)	Automatic Segmentation of recording in noisy channel	Knowledge-base Approach
(ÄŒernocký et al., 1998)	Dotch Speech Units	Knowledge-base & Data Driven
(Kannan and Ostendorf, 1998)	Comparison study	Knowledge-base
(Kessens et al., 1999)	Lexicon generation	Knowledge-base
Bacchiani (1999)	Continuous Speech Recognition	Knowledge-base Approach
Kessens (1999)	Added Pronunciation Variation To Lexicon To Improve Dutch Speech	Knowledge-base Approach
(Furuichi et al., 2000)	Phoneme Boundaries	Knowledge-base & Data Driven

Table 3-3: Recent Research Summary on English and Other Languages

Author	Purpose	Depends On
(Sekhar et al., 2003)	Classification of SWUs	Knowledge-Based
(Raju, 2003)	Language Identification	Knowledge-Based
(Jiang and Huang, 2003)	Analog to digital signal conversion and feature extraction methods	Knowledge-Based
(Mitchell et al., 2003)	Examine SWUs in decoding Tree	Knowledge-Based
(Adami and Hermansky, 2003)	Segment and label speech signal	Data Driven
(Kamakshi Prasad et al., 2004)	Automatic segmentation of continuous speech	Data Driven
(Sreenivas, 2004)	K-means segmentation	Data Driven
(Erdogan et al., 2005)	Rules to LM	Knowledge-Based+Data driven
(Sadohara et al., 2006)	Merge Similar Adjacent SWUs	Data Driven
(Walliczek et al., 2006)	SWUs	Knowledge-Based
(Hirsimäki et al., 2006)	Discovering Word Fragments	Data Driven
(Zhang and Nakamura, 2006)	Effective Phoneme Set	Data driven
(Rotovnik et al., 2007)	LVCSR for SLOVIAN Language (OOV-Huge)	Knowledge-Based
(Pan et al., 2007)	SWUs indexing	Knowledge-Based
(Tejedor et al., 2008)	Grapheme letter based SWUs	Knowledge-Based
(Kenney, 2008)	SWUs Determination	Data Driven
(Aksungurlu, 2008)	SWUs	Data driven
(Szoke et al., 2008)	Comparison between SWUs	Knowledge-Based

	methods	
(Majewski, 2008)	Syllable based	Knowledge-Based
(Rastrow et al., 2009)	Variable Length SWUs	Knowledge-Based+data driven
(Sarada et al., 2009)	Syllable like units	Data driven
(Choueiter, 2009)	Learning lexicon properties of OOV	Knowledge-Based
(Thangarajan et al., 2009)	Tamil Language Syllables	Knowledge-Based+data driven
(Rastrow et al., 2009)	Augmented word and SWUs	Data driven
(Zhang, 2009)	Keywords spotting	Data driven
(Pellegrini and Lamel, 2009)	Data driven word decomposition	Data driven
(Aimetti1 et al., 2010)	Computational Model To Learn SWUs	DD
(Iwami et al., 2010)	SWUs Recognition	Knowledge-Based
(Barroso et al., 2011)	SWUs selection	Knowledge-Based
(Baghdasaryan and Beex, 2011)	Phoneme Identification	Data Driven
(Duran et al., 2011)	Different Utterances Manifestations	Data Driven
(Venkatesh et al., 2011)	Limited Vocabulary , Isolated Word	Knowledge-Based
(Parada, 2011)	Transcribing OOV	Knowledge-Based
(Xie et al., 2011)	SWUs in Chines broadcaste stories	Data Driven
(Huijbregts et al., 2011)	Automatic Generation of SWUs	Data Driven
(Monica and Nagarajan, 2011)	Determines Phonemes	Data Driven
(Rahman et al., 2012)	Blind speech segmentation	Data Driven

CHAPTER 4

ARABIC SPEECH CORPUS AND STATISTICAL ANALYSIS OF ARABIC PHONEMES

4.1 INTRODUCTION

Sphinx Arabic Speech Recognizer uses 3-state HMM for each phone regardless of its length. A question arises here: whether this number of states is enough for some phonemes or should be a different number?

This chapter provides information for better modeling of the phonemes for speech recognition as well as for speech synthesis. The statistical information on the phoneme length can be used to enhance HMM modeling of phonemes, and for the design of appropriate neural network architecture for the phonemes. We studied various parameters, such as the number of frames a phoneme occupies, the phoneme frequency, and the mean length in frames, the standard deviation, the mode, and the median of the phoneme boundary. In addition, other language-model related information such as the bigram information is also studied. The results showed that phonemes can be clustered into groups based on a collection of statistical information that assist in design the most suitable HMM for each phoneme in terms of the number of states and other model parameters.

This Chapter is organized as follows: Section 2 describes the Arabic corpus used in the study Then, Section 3 is an explanation of the Arabic phoneme set. Section 4 describes the importance of the study. Section 5 overviewed the literature of phonemes statistical analysis. In section 6 we gave the details of our statistical analysis. We, finally conclude our chapter in Section 7.

4.2 Used Arabic Corpus

We have used a 5-hour recording of modern Arabic speech from different TV news clips, developed by King Fahd University of Petroleum & Minerals (KFUPM) with the support of King Abdulaziz City For Science And Technology (KACST) (Alghamdi et al. 2009). KACST is a governmental scientific organization representing Saudi Arabian national science agency and its national laboratories ("KACST," 2012). Part of its research support has been directed to Arabic speech recognition. The corpus is considered to be the main reference for modern standard Arabic speech recognition systems supported by KACST. The corpus consists of 16 KHz wav files of 10 millisecond frame duration; it contains 249 business/economic and sport stories (144 by male speakers and 105 by female speakers). The corpus wave file lengths range from 0.8 seconds to 15.6 seconds. An additional 0.1 second silence period is added at the beginning and at the end of each file. The corpus also includes the corresponding MFCC files and their transcriptions. The Arabic phonemes, Arabic dictionary and other script files used for manipulating corpus information are also included. It was used in the development of a continuous Arabic speech recognition system by (Ali et al., 2009).

4.3 The Arabic Phoneme Set

The Arabic phoneme set used in the corpus is shown in Table 4-1. Every phoneme and its corresponding English symbols are shown. The table also provides illustrative examples of the different vowel usages. This phoneme set is chosen based on the work related to Arabic text-to-Speech systems (Ahmed, 1991) and (Elshafei et al., 2002). The phoneme symbols were chosen in a way that are closely related to similar phoneme symbols used in the English ASR systems.

Table 4-1: Complete Arabic Phoneme List Used In Training (Ali, Elshafei, Al-Ghamdi, et al., 2008)

Phoneme	Arabic Letter	Example	Description
/AE/	ـَ	بَ	Short Vowel **FATHA**
/AE:/	ـَا	بَاب	Long Version of /AE/
/AA/	ـَ	خَ	Pharyngeal Version of /AE/
/AA:/	ـَ	خَاب	Long Version of /AA/
/AH/	ـَ	قَ	Emphatic Version of /AE/
/AH:/	ـَ	قَال	Long Version of /AH/
/UH/	ـُ	بُ	Short Vowel **DAMMA**
/UW/	ـُو	دُون	Long Version of /UH/
/UX/	ـُ	غُصن	Pharyngeal Version of /UH/
/IH/	ـِ	بِنت	Short Vowel **KASRA**
/IY/	ـِي	فِيل	Long Version of /IH/
/IX/	ـِ	صِنف	Pharyngeal Version of /IX/
/AW/	ـَو	لَوم	A Diphthong of both /AE/ and /UH/
/AY/	ـَي	صَيف	A Diphthong of both /AE/ and /IH/
/E/	ء	دفـء	Arabic Voiceless Glottal Stop **HAMZA**, and a variation for **QAF** in some dialects.
/B/	ب	حب	Arabic Voiced Bilabial Stop Consonant **BEH**
/T/	ت	توت	Arabic Voiceless Dental Stop Consonant **TEH**
/TH/	ث	دمث	Arabic Voiceless Inter-dental Fricative Consonant **THEH**
/JH/	ج	حج	Standard Arabic Voiced Palatal Stop Consonant **JEEM**, similar to English J.
/G/	ج	حجز	Egyptian Dialect (and others) for **JEEM.** Also used in foreign names. A Velar version of /JH/.
/ZH/	ج	جايز	A Voiced Fricative Version of /JH/
/HH/	ح	مح	Arabic Voiceless Pharyngeal Fricative Consonant **HAH**
/KH/	خ	خادم	Arabic Voiceless Pharyngeal Velar Consonant **KHAH**
/D/	د	مد	Arabic Voiced Dental Stop Consonant **DAL**
/DH/	ذ	فذ	Arabic Voiced Inter-dental Fricative Consonant **THAL**
/R/	ر	حر	Arabic Dental Trill Consonant **REH**
/Z/	ز	حز	Arabic Voiced Dental Fricative Consonant **ZAIN**, and a variation of **THAL** in many dialects.
/S/	س	شمس	Arabic Voiceless Dental Fricative Consonant **SEEN**
/SH/	ش	دش	Arabic Voiceless Palatal Fricative Consonant **SHEEN**
/SS/	ص	فحص	Arabic Emphatic Voiceless Dental Fricative Consonant **SAD**

/DD/	ض	محض	Arabic Emphatic Voiced Dental Stop Consonant **DAD**
/TT/	ط	مط	Arabic Emphatic Voiceless Dental Stop Consonant **TAH**
/DH2/	ظ	حفظ	Arabic Emphatic Voiced Dental Fricative Consonant **THAH**
/AI/	ع	مع	Arabic Voiced Pharyngeal Fricative Consonant **AIN**
/GH/	غ	لدغ	Arabic Voiced Velar Fricative Consonant **GHAIN**, also a variation of **QAF** in many dialects.
/F/	ف	صنف	Arabic Voiceless Labial Fricative Consonant **FEH**
/V/	-		Voiced Version of **FEH**. Exists in Foreign Names Only.
/Q/	ق	دق	Arabic Voiceless Uvular Stop Consonant **QAF**
/K/	ك	صك	Arabic Voiceless Velar Stop Consonant **KAF**
/L/	ل	مصل	Arabic Approximant Dental Consonant **LAM**
/M/	م	مدام	Arabic Nasal Labial Consonant **MEEM**
/N/	ن	جنون	Arabic Nasal Dental Consonant **NOON**
/H/	هـ	صه	Arabic Voiceless Glottal Fricative Consonant **HEH**
/W/	و	نرجو	Arabic Velar Approximant Semi-vowel **WAW**
/Y/	ي	وادي	Arabic Palatal Approximant Semi-vowel **Yeh**

From the table in Table 4-1, we can see that, the regular Arabic short vowels /AE/, /IH/, and /UH/ correspond to the Arabic diacritical marks Fatha, Kasra, andDamma , respectively. “The /AA/ is the pharyngealized allophone of /AE/, which appears after an emphatic letter. Similarly, the /IX/ and /UX/ are the pharyngealized allophones of /IH/ and /UH/ respectively. When /AE/ appears before an emphatic phoneme, its allophone /AH/ is used instead. When a short vowel is located between two nasal phonemes in the same syllable, it is likely to be nasalized. The allophones /AN/, /IN/, and /UN/ are the nasalized versions of /AE/, /IH/ and /UH/, respectively. The regular Arabic long vowel phonemes are /AE: /, /IY/ and /UW/, respectively. The length of a long vowel is normally equal to two short vowels. The phonemes /AY/ and /AW/ are actually two vowel sounds in which the articulators move from one post to another.

These vowels are called Diphthongs. The allophone /AY/ appears when a Fatha comes before an undiacritized Yeh. Similarly, /AW/ appears when a Fatha comes before an undiacritized Waw. The Arabic voiced stops phonemes /B/ and /D/ are similar to their English counter parts. /DD/ corresponds to the sound of the Arabic Dhad letter. The Arabic voiceless stops /T/ and /K/ are basically similar to their English counter parts.

The sound of the Arabic emphatic letter Qaf is represented by the phone /Q/. The Hamza plosive sound is represented by the phone /E/, and the sound of Jeem (in many dialects) is represented by /G/. The voiceless fricatives are produced with no vibration of the vocal cords/folds. The sound is produced by the turbulence flow of air through a constriction.

The Arabic voiceless fricatives /F/, /S/, /TH/, /SH/, and /H/ are basically similar to their English twins. In addition, the Arabic phones /SS/, /HH/, and /KH/ are the sounds of the Arabic letters Sad, Hah and Khah, respectively.

Voiced Fricatives are generated with simultaneous vibration of the vocal cords. The Arabic voiced fricative phones are /AI/, /GH/, /Z/, and /DH/ corresponding to the sound of the Arabic letters: Ain, Ghain, Zain, and Thal. The Arabic affricate sound /JH/ is similar to the corresponding one in English, while /ZH/ is a concatenation of a voiced stop followed by a fricative sound. The Arabic resonant are similar to the English resonant phonemes. These are /Y/ for Yeh, /W/ for Waw, /L/ for Lam, and /R/ for Reh" (Ali, Elshafei, Al-Ghamdi, et al., 2008).

4.4 The Need of Arabic Phoneme Statistics

A thorough statistical analysis of currently used Arabic phonemes from a widely used Arabic corpus is presented in this chapter. The aim of this study was to understand the abstract concept and behavior of the Arabic phonemes and to investigate the average number of frames that each phoneme occupies. The main objective was to determine the suggested number of frames that each SWU initially should take. In addition, it was an attempt to increase the speed of recognition by reducing the HMM chain.

As we have seen in CHAPTER 3, most of the research work carried out on continuous Arabic speech recognition utilizes Hidden Markov Models (HMM) achieving different rates of recognition accuracy (Al-Zabibi, 1990) (Ali et al., 2009). Phonemes recognition accuracy is measured by the percentage of correctly recognized phonemes. The ASR accuracy using HMMs is affected by several factors, including the presence of noise; the phoneme set used; the number of HMM states allocated for each phoneme and the duration of each one. Improving the performance of current ASR techniques requires thorough investigation of these factors in order to determine the areas of improvement. However, no statistical analysis at the phoneme level has been performed, to our best knowledge, on the currently used Arabic speech phonemes.

Statistical analysis of Arabic phonemes provides a clear pictorial view of phonemes behavior and gives the ability to control this behavior by analyzing the collected statistics. For example; knowing the frequency of a phoneme in a suitable Arabic corpus can be used to replace a misrecognized phoneme in an utterance by the most probable one. The probability of occurrence of a phoneme is calculated by using the following equation (Walpole et al., 2007) :

$$\mathbf{P(Phoneme)} = \frac{\textbf{Frequency of Phoneme}}{\textbf{Total Number Of Phonemes}} \tag{4-1}$$

Moreover, knowing the average duration of a phoneme in terms of the number of frames it occupies can be used to determine the number and configuration of HMM states that are most suitable for recognizing it. Other statistical information like phonemes duration distributions, mode, median, and standard deviation are helpful in making the right decision concerning unrecognized phonemes during the recognition process.

In Section 4.6, we introduce a full statistical analysis of Arabic phonemes that can be used to improve continuous ASR accuracy by reducing word error rate (WER) (Jurafsky and Martin, 2000). The word error rate is computed by the following formula:

$$\mathbf{WER} = \frac{\mathbf{S+D+I}}{\mathbf{S+D+C}} \tag{4-2}$$

Where *S* is the number of substitutions, *D* is the number of deletions, *I* is the number of insertions and *C* is the number of corrections.

We carry out this analysis on the Arabic speech corpus that was developed in Saudi Arabia by KFUPM with the support of King Abdul-Aziz City for Science and Technology (KACST) (See section 4.2).

4.5 Statistical Analysis in the Literature

Multiple research work has been carried out regarding statistical analysis of phonemes in languages other than Arabic. A short report which investigated the relationship between population size and phoneme inventory size based on statistical analysis methods was presented by (Hay and Bauer, 2007) . A robust correlation between the two is found where the language that have more speakers, it is likely to have bigger phoneme inventory (Hay and Bauer, 2007) . The results of (Hay and Bauer, 2007) were independent of the language and hold for both vowel inventories and consonant inventories.

A statistical procedure based on Fast Fourier transformation (FFT) to classify word-initial voiceless obstruent was presented by (Forrest, K. et al., 1988). Each FFT was treated as a random probability distribution from which the first four moments (mean, variance, skewness, and kurtosis). The model constructed from the males' data correctly classified about 94% of the voiceless stops produced by the female speakers by (Forrest, K. et al., 1988). Classification of the voiceless fricatives did not exceed 80%.

A technique for determining the pure speech-signal in a noisy environment and some statistical analysis methods for phoneme-isolation was introduced by (Tak and Bhargava, 2010). Relatively high accuracy of phoneme isolation is achieved in noisy environment. Moreover, the quality of phoneme separation was improved based on this statistical analysis.

(Yu et al., 2011) statistically studied the degree of contrast between Chinese phonemes. Results from (Yu et al., 2011) showed that the high degree of sparsity and the low degree of contrast of human languages not only leave enough room for new words, new dialects and new languages to appear, but also contribute to effective and reliable communication. The presence of few phonemic mistakes is unlikely to cause wrong decoding (sound recognition) and/or failed communication.

Punjabi syllables from a large Punjabi corpus were statistically analyzed in (Singh and Lehal, 2010). To minimize the database size, efforts were directed toward the selection of a minimal set of syllables covering almost the whole Punjabi word set (Singh and Lehal, 2010). The corpus contains more than 104 million words. The authors reported that the analysis resulted in a relatively smaller syllable set. Those results produced an improved text-to-speech system and enhanced the speech recognition, raising its accuracy, as per the authors report..

Studies on Arabic language statistical phoneme analysis are rare in the literature. However, we mention some work that is closely related to the subject. Statistical studies on 10,000 Arabic words (converted to phonemic form) involving different combinations of broad phonetic classes were carried out in (Al-Zabibi, 1990). Some particular features of the Arabic language had been exploited. The results showed that vowels represent about 43% of the total number of phonemes. He also showed that about 38% of the words could be uniquely represented using eight broad phonetic classes. When introducing detailed vowel identification, the percentage of uniquely specified words increased to 83%. (Al-Zabibi, 1990) suggested that, a fully detailed phonetic analysis of the speech signal is unnecessary.

Some phonemes confusability in modern Arabic standard speech is resolved by (Maalyl and Ahmed, 2002). Data mining techniques, spectral analysis and some statistical modeling are used to resolve the problem of confusability between phonemes that are very close to each other in pronunciation (For example, the confusion encountered between /H/ (haa ـه) and /E/ (hamzah ءْ)). The statistical analysis done by (Maalyl and Ahmed, 2002) was limited as compared to our analysis.

4.6 Detailed Statistics

Our workflow methodology consists of a sequence of five phases shown in Figure 4-1. The phases are: Corpus Acquisition, Phoneme Segmentation, Data Processing, Information Gathering and Statistical Analysis.

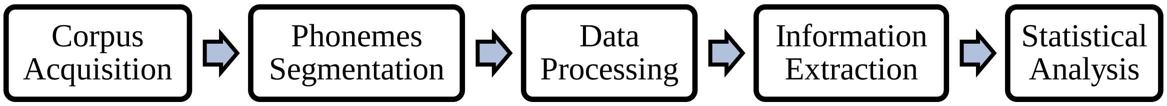


Figure 4-1: Workflow phases used to perform the statistical analysis.

4.6.1 Phoneme Segmentation

We feed the corpus as to the CMU sphinx speech recognition engine to produce the phonemes and words segmentation of all utterances using the trigram method. Table 4-2 shows a sample of both the word segmentation and the phonemes. The MFCC files (features) of the corpus and their corresponding transcriptions were fed to CMU Sphinx speech engine in order to apply forced alignment to generate the phoneme and word segmentation for all utterances.

Table 4-2: Samples of Word Segmentation (Left), Part of Phoneme Segmentation (Right) (The word ارتفعت).

Word Segmentation			Phoneme Segmentation for the word (اِرتَفَعَت)				
Start Frame	End Frame	Word	Start Frame	End Frame	Phone	Previous Phone	Next Phone
0	12	<s>	0	12	SIL		
13	15	<s>	13	15	SIL		
16	59	اِرتَفَعَت	16	21	E	SIL	IH
60	101	الأسهُم	22	24	IH	E	R
102	162	الكُوَيتِيَّة	25	27	R	IH	T
163	206	اِرتِفَاعَا (٢)	28	30	T	R	AE
207	262	طَفِيفًا	31	33	AE	T	F
263	292	اليَوم	34	36	F	AE	AE
293	332	السَّبت	37	39	AE	F	AI
333	341	</s>	40	44	AI	AE	AE
342	348	</s>	45	51	AE	AI	T
			52	59	T	AE	E
			60	65	E	T	L

4.6.2 Data Processing and Information Extraction

For the purpose of extracting useful information from the previous generated files, a MATLAB program was used to compute the frequency, lengths, mean lengths, standard deviation of lengths, the distribution of means, the distribution of the lengths of each phoneme, mode of the length for each phoneme and the median of the lengths. Moreover, we extracted the length probability occurrence of every phoneme. These statistics are shown in Table 4-3. In Table 4-3, also shows the corresponding English representation of every Arabic phoneme.

Figure 4-2 shows the mean of phonemes lengths distribution measured in frames, and the frequency of each phoneme in the whole 5 hour corpus. The distributions of phoneme lengths were also produced.

Table 4-3: Arabic Phonemes Statistics (For Arabic Phonemes Meanings See Table 4-1)

Arabic Phonemes	English Format	Frequency	Probability	MIN-Length	Max-Length	Mean	STD	Mode	Median
صمت	SIL	13676	0.0626	3	30	8.23	3.9	3	8
تشويش	NIOSE	71	0.0003	3	30	22.76	16.47	3	19
ـَ	AA:	1730	0.0079	3	30	8.28	3.23	7	8
ـَ	AH:	667	0.003	3	29	8.23	2.82	7	8
ـِ	IH	21199	0.097	3	30	5.32	2.82	3	5
ـِي	IY	4603	0.021	3	30	8.57	6.2	4	7
ـِ	IX	843	0.0039	3	30	8.77	6.05	7	8
نفس	+INH+	347	0.0016	3	30	14.04	11.24	4	10
ـَ	AE	28806	0.1319	3	30	5.66	2.92	4	5
ـَ	AA	3606	0.0165	3	30	5.71	3.45	5	5
ـَ	AH	1096	0.005	3	20	6.19	2.48	5	6
ـُ	UH	6164	0.0282	3	30	5.66	3.17	4	5
ـُو	UW	2669	0.0122	3	30	8.03	4.56	6	7
ـُ	UX	2356	0.0107	3	26	5.38	2.28	3	5
ـَو	AW	607	0.0027	3	29	10.04	3.6	9	10
ء	E	12742	0.0583	3	30	5.71	4.12	3	4
ـَى	AY	835	0.0038	3	30	11	4.26	9	10
ب	B	4735	0.0216	3	30	6.64	4.29	5	6
ت	T	9995	0.0457	3	30	7.61	3.77	7	7
ث	TH	1229	0.0056	3	30	9.33	3.2	9	9
ج	JH	1742	0.0079	3	30	9.12	4.22	8	9
ح	HH	2011	0.0092	3	30	9.98	3.92	10	10
خ	KH	1329	0.006	3	30	9.91	3.38	9	10
د	D	4455	0.0203	3	30	6.88	4.47	5	6
ذ	DH	680	0.0031	3	30	6.09	2.6	5	6
ر	R	7671	0.0351	3	30	6.27	5.51	4	5
ز	Z	1040	0.0047	3	30	9.8	4.99	9	9
س	S	4079	0.0186	3	30	9.73	3.36	9	10
ص	SS	1353	0.0061	3	30	10.66	3.44	9	10
ش	SH	1658	0.0075	3	30	10.98	3.15	10	11
ض	DD	867	0.0039	3	30	7.04	4.19	6	6
ط	TT	1354	0.0062	3	30	8.28	4.2	7	7
ظ	DH2	227	0.001	3	18	6.8	2.21	6	6
ع	AI	5138	0.0235	3	30	8.31	4.45	7	8
غ	GH	485	0.0022	3	30	7.63	4.07	7	7
ف	F	3895	0.0178	3	30	8.47	4.72	8	8
ق	Q	3060	0.014	3	30	9.66	3.67	9	9
ك	K	2505	0.0114	3	30	9.28	4.52	9	9
ل	L	14471	0.0662	3	30	5.54	3.64	4	5
م	M	9669	0.0442	3	30	6.71	4.57	5	6

Arabic Phonemes	English Format	Frequency	Probability	MIN-Length	Max-Length	Mean	STD	Mode	Median
ن	N	9471	0.0433	3	30	6.85	5.93	4	5
هـ	H	5200	0.0238	3	30	8.82	8.53	3	6
و	W	4420	0.0202	3	30	8.48	5.88	6	7
ي	Y	5149	0.0235	3	30	7.8	6.12	6	7
ـَا	AE:	8480	0.0388	3	30	8.12	4.59	6	7

Figure 4-3 shows a sample of 20 phonemes distributions. In order to further analyze the properties of the Arabic phonemes, some graphs are depicted in Figure 4-4,

Figure 4-5, Figure 4-6 and Figure 4-4. Figure 4-4, Shows the occurrence probability distribution of the Arabic phonemes in the corpus. We have to note that, the silent, noise and inhalation phonemes are not part of speech even though they are included in the graph. An interesting result from

Figure 4-5 shows that phonemes with similar means form clusters in the phoneme set. We postulate that these clusters will improve phonemes recognition as we will see in the analysis section.

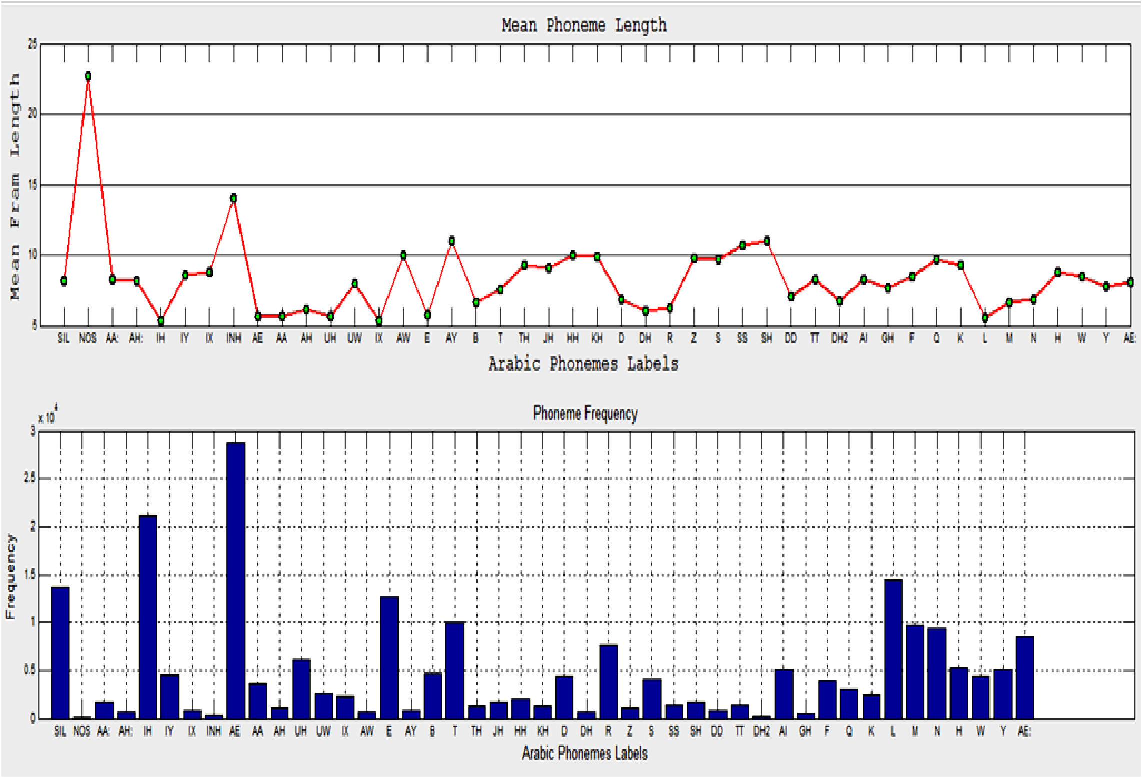


Figure 4-2: (Upper Graph): Arabic Phoneme Mean Length Distribution. (Lower Graph): Arabic Phoneme Frequency Distribution.

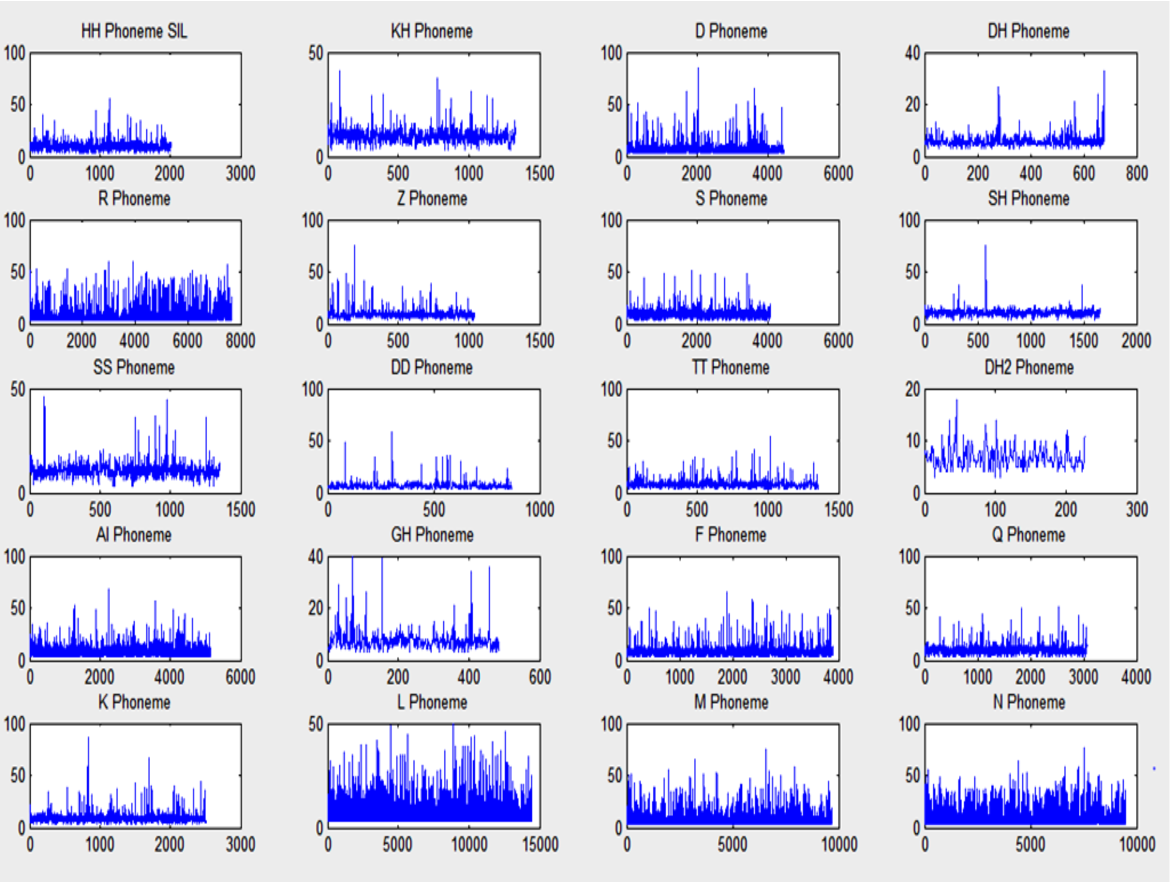


Figure 4-3: Phoneme length distributions for selected 20 Arabic Phonemes

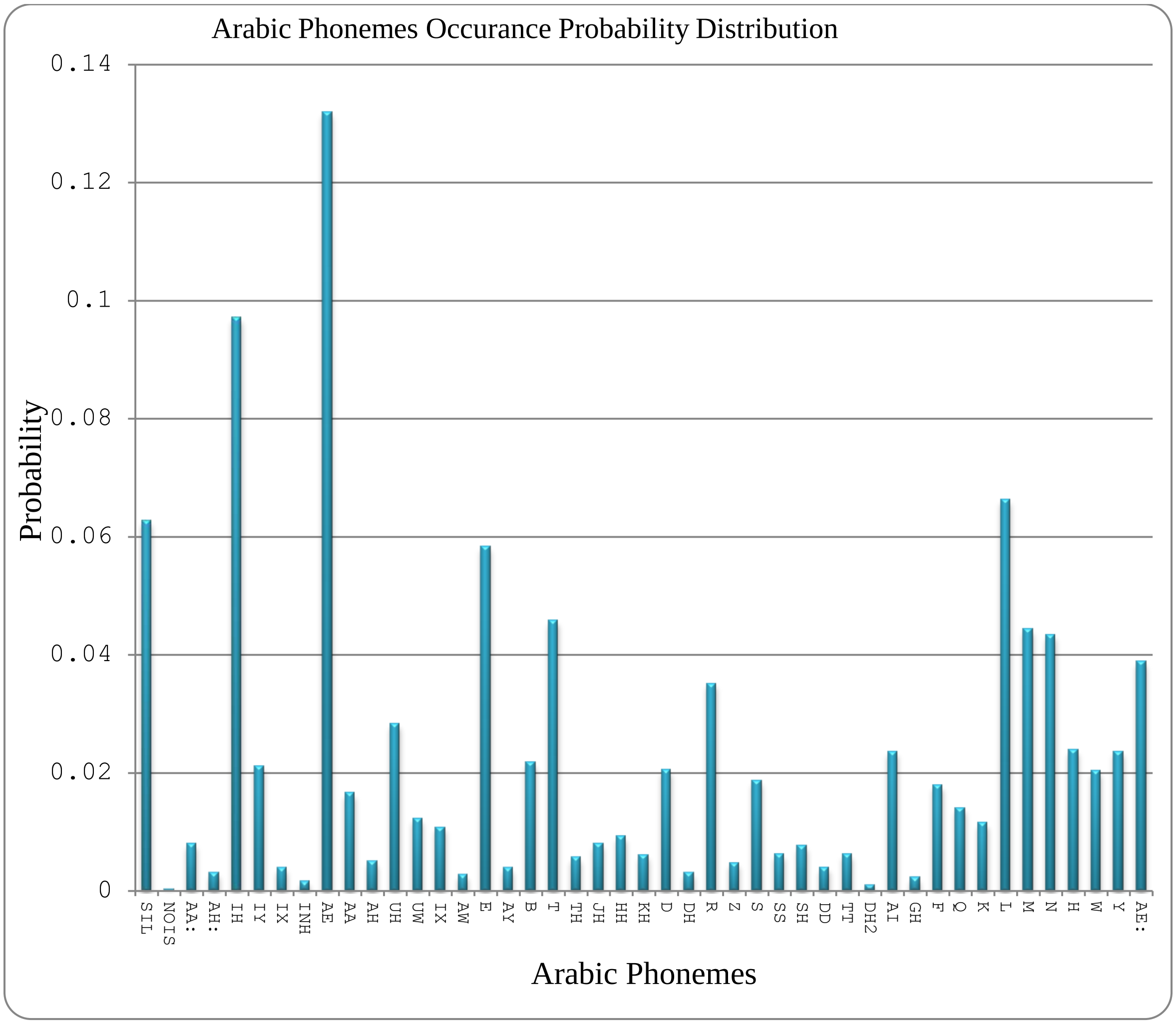


Figure 4-4: Phoneme Probability Distribution over the Population

The median values of the phoneme length clarify this clustering. Classes of the phonemes are being distinguished from each other and a clear separation between phoneme classes becomes more evident, as shown in Figure 4-6 another important graph that can be studied is the one highlighting the most frequent length of all occurrences of a certain phoneme, appearing in the corpus. This is called the mode, and is shown in Figure 4-7

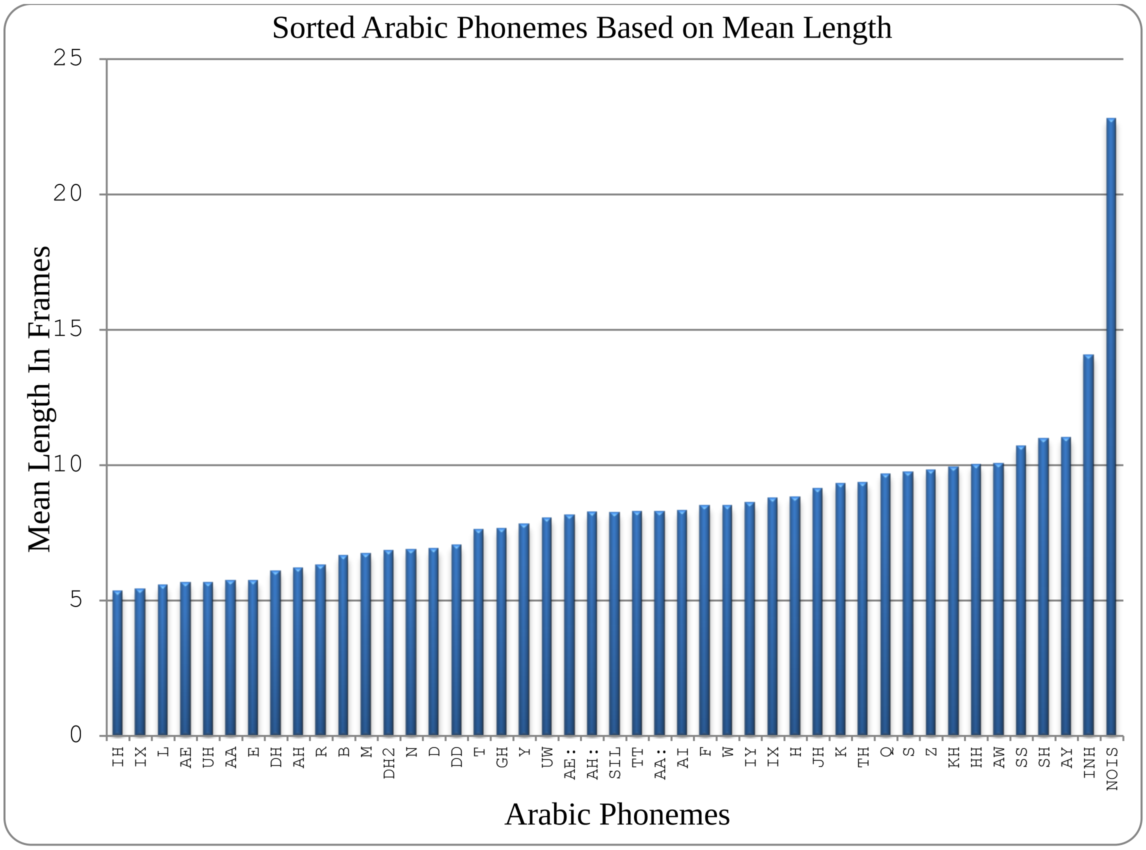


Figure 4-5: Arabic Phonemes Based on Their Mean Lengths

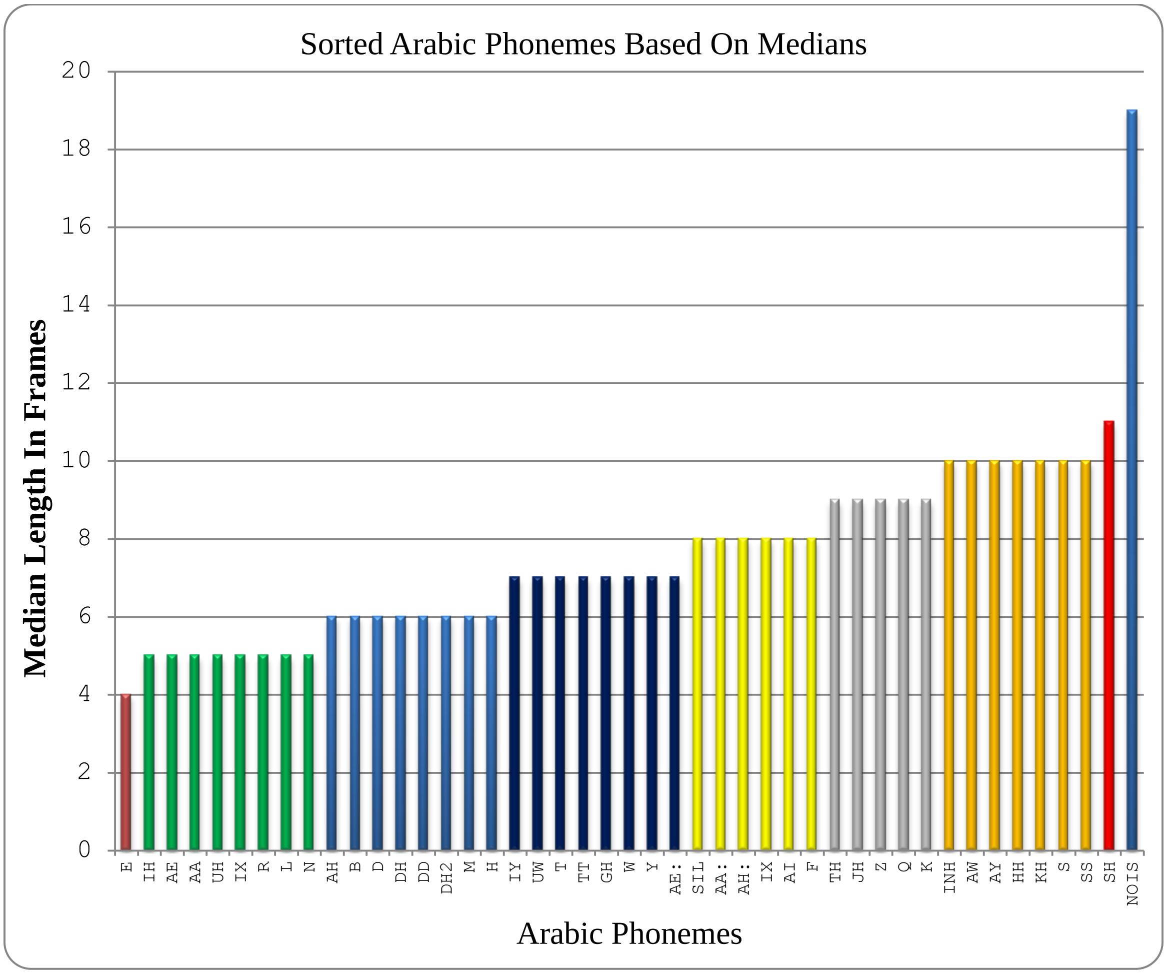


Figure 4-6: Arabic Phonemes Based on Their Median Length.

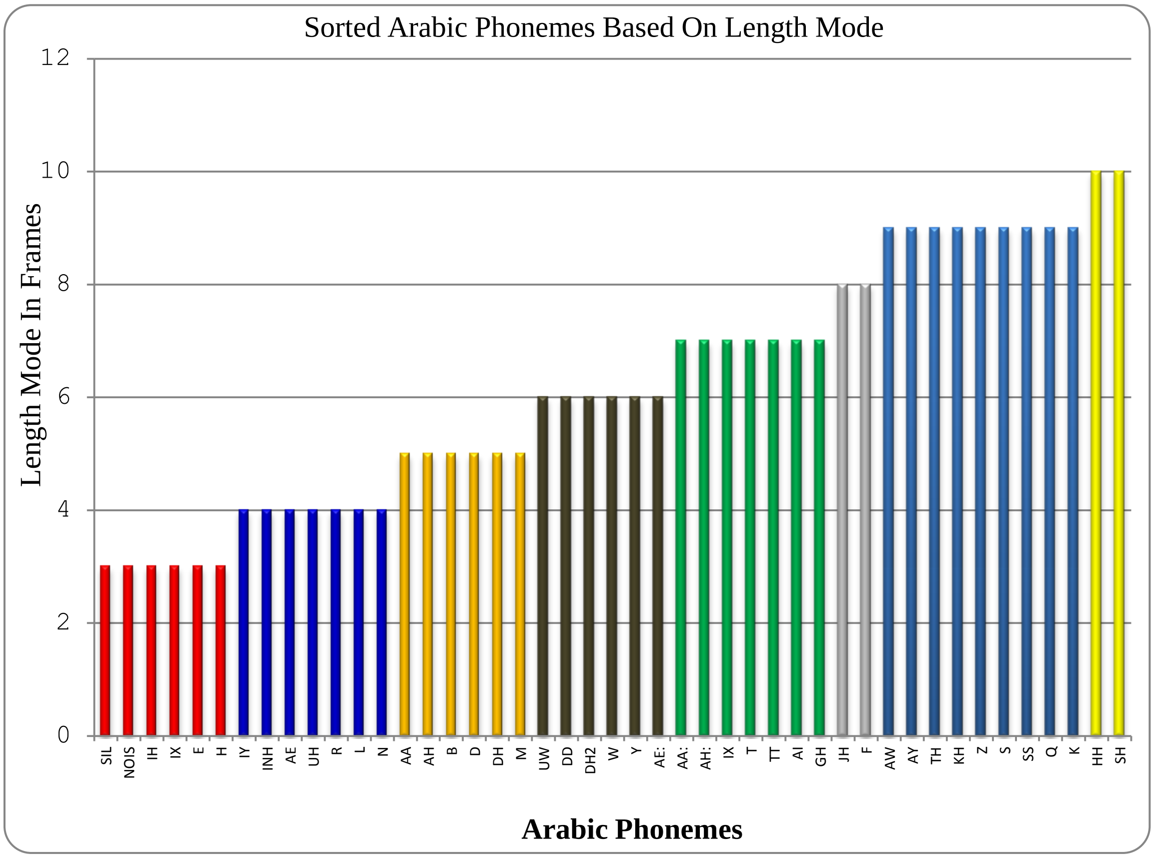


Figure 4-7: Arabic Phonemes Based on Their Mode Length.

Table 4-4, shows samples of the bigrams of three phonemes. It shows the probability that a phoneme comes after another specific phoneme. The tri-gram frequency for each phoneme is also produced, which represents the frequency that the phoneme occurs as one of triple phonemes. Table 4-5 shows the trigram frequency of each phoneme. The bigram table (Table 4-4) was generated by the HTK tool, with the probability being calculated according to the following Equation:

$$\mathbf{P}(\mathbf{i},\mathbf{j}) = \begin{cases} \frac{\mathbf{N}(\mathbf{i},\mathbf{j})-\mathbf{D}}{\mathbf{N}(\mathbf{i})} & \textbf{if } \mathbf{N}(\mathbf{i},\mathbf{j}) > \mathbf{t} \\ \mathbf{b}(\mathbf{i})\mathbf{p}(\mathbf{j}) & \textbf{Otherwise} \end{cases} \qquad (4\text{-}3)$$

Here, N(i,j) is the number of times word (j) follows word (i), and N(i) is the number of times word (i) appears, and b(i) is the bigram count. A small part of the available probability mass is deducted from higher bigrams. This process is called discounting; the symbol (D) represents this value, it is usually 0.5. When a bigram count falls below the threshold (t), the bigram is backed-off to the unigram probability suitably scaled by a back-off weight in order to ensure that all bigram probabilities for a given history sum to one (Young et al., 2006). Note that Table 4-4 shows the base-10 log of the probability values computed.

Table 4-4: Bigram Tables for Phonemes DD, DH, and DH2 (Prob. here is the Log base 10 Probability).

Prob	Phone	Next	Prob	Phone	Next	Prob	Phone	Next
-2.394	DD	+INH+	-2.6564	DH	+INH+	-2.6571	DH2	+INH+
-0.5323	DD	AH	-0.6137	DH	AE	-2.6571	DH2	AE
-0.8499	DD	AH:	-0.6163	DH	AE:	-0.2934	DH2	AH
-1.7205	DD	AI	-2.4346	DH	AI	-1.0236	DH2	AH:
-2.5401	DD	AW	-1.9031	DH	B	-2.6571	DH2	E
-2.1251	DD	AY	-2.4346	DH	E	-1.3348	DH2	H
-1.8077	DD	E	-2.4346	DH	F	-1.066	DH2	IH
-2.394	DD	F	-2.6564	DH	H	-0.9012	DH2	IY
-2.1977	DD	HH	-2.6564	DH	HH	-2.6571	DH2	M
-0.6513	DD	IH	-0.7852	DH	IH	-2.1799	DH2	R
-0.9066	DD	IY	-0.7136	DH	IY	-1.066	DH2	UH
-1.9603	DD	L	-1.5895	DH	K	-1.9581	DH2	UW
-2.394	DD	M	-2.6564	DH	L			
-2.394	DD	R	-3.1335	DH	M			
-2.5401	DD	S	-2.6564	DH	Q			
-1.7767	DD	SIL	-3.1335	DH	SH			
-2.7619	DD	T	-3.1335	DH	SIL			
-1.1563	DD	UH	-2.4346	DH	T			
-1.5858	DD	UW	-1.1841	DH	UH			
-1.9603	DD	W	-3.1335	DH	UW			
			-2.6564	DH	W			
			-2.1793	DH	Y			

Table 4-5 : Number of Triphones for Each Phoneme (Abuzeina et al., 2011)

Arabic Phoneme	Triphones	Arabic Phoneme	Triphones	Arabic Phoneme	Triphones
AA	96	L	560	IH	657
AA:	70	M	344	IX	85
AE	542	N	454	IX:	51
AE:	389	Q	238	IY	372
AH	64	R	460	JH	181
AH:	40	S	302	K	225
AI	289	SH	144	KH	130
AW	77	SS	156	IH	657
AY	104	T	393		
B	324	TH	106		
D	356	TT	161		
DD	137	UH	487		
DH	65	UW	257		
DH2	41	UX	70		
E	479	W	187		
F	286	Y	218		
GH	83	Z	192		
H	258	HH	195		

A triphones is simply a group of 3 phones in the form "L-X+R" - where the "L" phone (i.e. the left-hand phone) precedes "X" phone and the "R" phone (i.e. the right-hand phone) follow it. Figure 4-8 represents the triphone frequencies of the phonemes.

Another important probability is the probability that reflects the phoneme occurrences in a specific length. We found that all Arabic phoneme lengths are distributed between 3 and 30 frames. Table 4-6 shows selected samples of these probabilities.

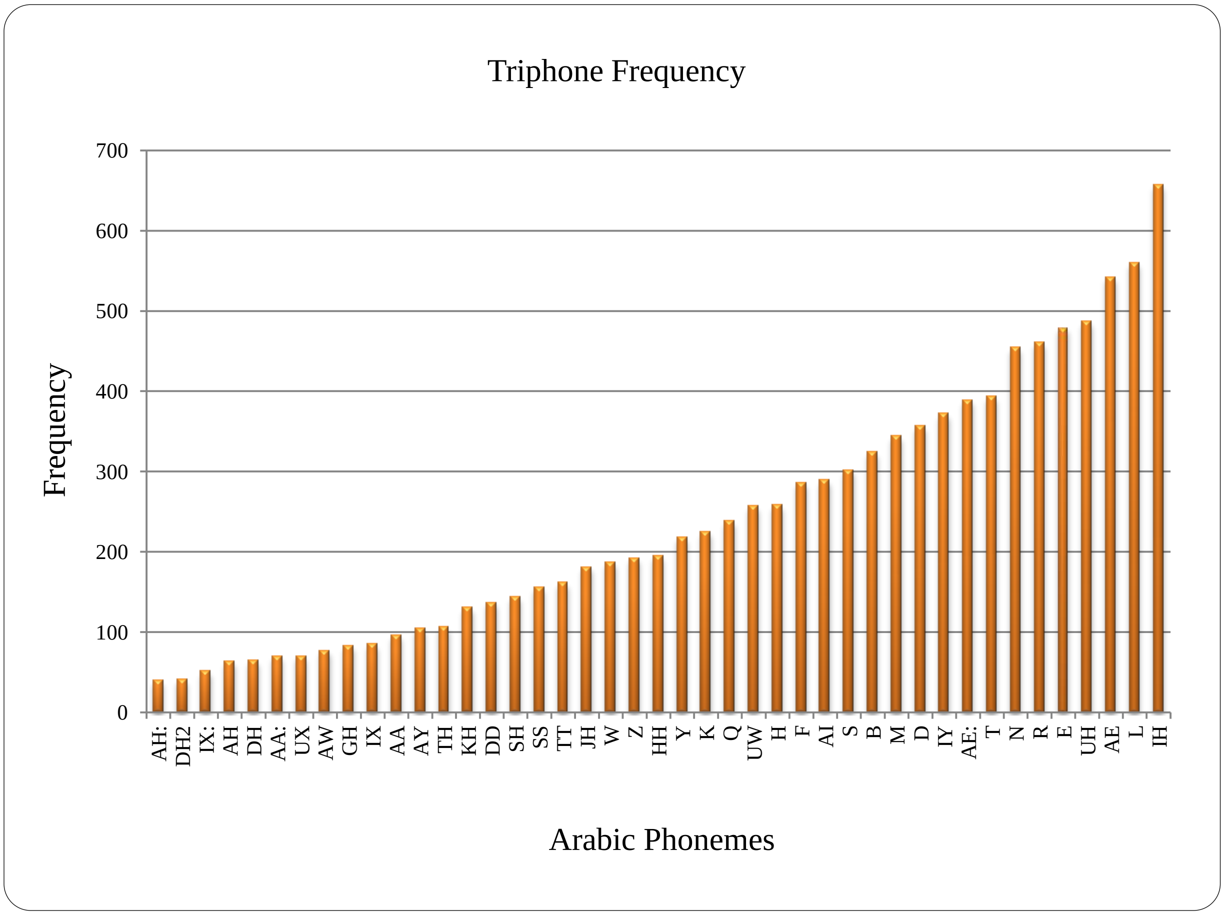


Figure 4-8: Triphone Frequencies of Arabic Phonemes

Table 4-6: Sample of Phonemes Length Probability

Length	Probability						
	AA:	AH:	IH	IY	IX:	AE	AA
3	0.0295	0.0225	0.2286	0.1032	0.0522	0.1496	0.1534
4	0.0416	0.0435	0.2271	0.1186	0.0842	0.2410	0.2188
5	0.0960	0.0810	0.1904	0.0925	0.1091	0.2235	0.2432
6	0.1393	0.0945	0.1414	0.0967	0.1210	0.1548	0.1606
7	0.1578	0.1814	0.0861	0.1086	0.1293	0.0850	0.0879
8	0.1410	0.1694	0.0492	0.1082	0.1186	0.0491	0.0466
9	0.1052	0.1439	0.0267	0.0862	0.0913	0.0251	0.0208
10	0.0879	0.1109	0.0157	0.0652	0.0652	0.0165	0.0161
11	0.0607	0.0540	0.0092	0.0450	0.0558	0.0109	0.0086
12	0.0555	0.0315	0.0060	0.0363	0.0474	0.0092	0.0078
13	0.0150	0.0255	0.0047	0.0215	0.0273	0.0075	0.0075
14	0.0260	0.0150	0.0029	0.0224	0.0154	0.0059	0.0067
15	0.0133	0.0045	0.0019	0.0167	0.0107	0.0055	0.0033
16	0.0127	0.0090	0.0013	0.0128	0.0178	0.0036	0.0042
17	0.0069	0.0060	0.0017	0.0102	0.0083	0.0032	0.0039
18	0.0040	0.0015	0.0010	0.0065	0.0059	0.0019	0.0014
19	0.0040	0.0030	0.0008	0.0052	0.0000	0.0015	0.0022
20	0.0006	0.0015	0.0006	0.0048	0.0071	0.0011	0.0008
21	0.0006	0.0000	0.0005	0.0043	0.0012	0.0010	0.0003
22	0.0006	0.0000	0.0004	0.0041	0.0024	0.0011	0.0011
23	0.0006	0.0000	0.0006	0.0024	0.0012	0.0007	0.0003
24	0.0000	0.0000	0.0004	0.0009	0.0012	0.0005	0.0003
25	0.0000	0.0000	0.0002	0.0026	0.0024	0.0004	0.0006
26	0.0000	0.0000	0.0002	0.0030	0.0024	0.0002	0.0000
27	0.0006	0.0000	0.0003	0.0011	0.0047	0.0003	0.0003
28	0.0000	0.0000	0.0004	0.0020	0.0000	0.0002	0.0006
29	0.0000	0.0015	0.0001	0.0017	0.0024	0.0001	0.0003
30	0.0000	0.0000	0.0001	0.0022	0.0012	0.0001	0.0003

4.6.3 Statistical Analysis

By looking at the previous graphs, tables and based on the 5-hour corpus, we find that phonemes occur in different frequencies, with the top most frequent phonemes being AE (ـَ), L (ل) and IH (ـِ) respectively, while the least frequent phonemes are GH (غ), DH2 (ظ) and DH (ض), respectively, ignoring INH (Inhalation تنفس), NOISE.(تشويش) and SIL (صمت). As we see from the graph the phoneme /AE/ is the most probable phoneme when a phoneme is missed in recognition (as it is shown in

Figure 4-2 and Figure 4-4). In general, Figure 4-4 can be used whenever confusion or misrecognition of a phoneme happens during the recognition phase. This, in turn, can raise the probability of achieving higher accuracy.

Figure 4-3 gives a sample of 20 phonemes length distributions from the set of Arabic phonemes. From this Figure, we can see the frequency and behavior of the phoneme length through the corpus. The Arabic phoneme mean length distribution gives a clear idea about the average length of each phoneme. A closer look into the graphs reveals that we can cluster phonemes with similar average lengths together. For example the phonemes /IX/ (ـِ) and /IH/ (ـِ) are in the first cluster, based on the average length in frames. Another cluster is the phonemes /UH/ (ـُ), /AA/ (ـَ) like in (خَ) and /E/ (ء) as seen from

Figure 4-5. Knowing the phoneme average length, in general, will enhance the Arabic speech recognition by limiting the number of HMM states that could be used for each phoneme. Moreover, if a specific phoneme is missing from the recognition process and a gap of the recognition of a specific length occurs then; this gap could be filled with

one of the phonemes that belong to same gap length cluster. In Figure 4-6, a clear border have emerged between the phoneme clusters. We notice that /E/ (ء) was seen to be one cluster while other phonemes are grouped together, ignoring /+INH+/ (Inhalation تنفس), /NOISE/ (تشويش) and /SIL/ (صمت).

Figure 4-7, also shows that the mode of each phoneme can be used to recognized missed phonemes by replacing them with the closest mode phoneme. Table 4-4 presents the bigram language model information. This model is usually used to enhance the recognition rate by replacing missing or improbable phonemes with the maximum likelihood phoneme. Such information can be used with other statistical information, like the mode and the cluster, to further enhance the accuracy. Table 4-5 shows the count of phonemes observed in the middle location of two other phonemes in the corpus. For example, the phoneme (E ء) mostly occurs between two phonemes since the table shows 479 occurrences of the phoneme in the middle. However, the phoneme (AH: ـَ fatha) is rarely located in between two phonemes. During recognition and if a phoneme is missed at the middle, the triphone frequencies can be utilized to determine the most appropriate phoneme based on Table 4-4 with highest probability of likelihood. Figure 4-8 depicts the triphone frequencies of a selected set of phonemes from the corpus. Finally, Table 4-6 presents information that can be used in determining the number of HMM states for each phoneme based on the maximum probability of its length occurrence.

4.7 Chapter Summary

The statistical study and analysis of the Arabic phonemes are necessary to improve the performance of current Arabic ASR systems. Knowing the length of a specific phoneme can be used to choose the length of the HMM chain that represents it, which in turn increases the speed of recognition and its accuracy. Clustering the phonemes based on the median of the lengths of each one can help narrowing the search for the suitable phoneme during the recognition phase, which also increases the speed and accuracy of the ASR. Statistical analysis can also be useful in developing other methods for phonemes segmentation. Language model information, such as bigrams, can be used to replace multiple equi-probable phonemes with the correct ones, and hence reduce the word error rate.

Some major finding that we noted that; the most frequent phonemes were AE (short vowel fataha), L (ل) and IH (short vowel kasrah) respectively and the least frequent phonemes were GH (غ), DH2 (ظ) and DH (ض) respectively. Based on these results we can say AE is the most probable phoneme. Moreover, we notice that E (ء) was seen to be one cluster while other phonemes were grouped together. The phoneme (E ء) (الهمزة الساكنة) mostly occur between two phonemes because from the table, 479 times it is located at the middle while phoneme (AH: long vowel fataha like in قَ) is rarely located at the middle of two phonemes.

More investigation is needed in the future to check the effect of our statistical analysis on the recognition rate and to determine the best configuration for building the acoustic model. It is also necessary to carry out this analysis on other Arabic corpuses and compare results.

CHAPTER 5

ARABIC PHONEME TRANSCRIPTION USING DATA DRIVEN APPROACH

5.1 INTRODUCTION

Phoneme transcription plays an important role in the process of speech recognition, text-to-speech applications, and speech database construction (Kim et al., 2005). Two traditional methods are usually used for phoneme transcription; the feature input method, which is carried out by the speech recognition task, and the text input method, which is carried out by Grapheme to Phoneme task (G2P) (Liang et al., 2006). The maximum accuracy reached in continuous speech recognition with large vocabulary was 80% (Liang et al., 2006). G2P gives more accurate recognition, but relies on a perfect pronunciation lexicon.

Phonemes recognition in continuous speech is not an easy task due to coarticulation, and the inherent variability in the pronunciation of some phonemes. Stop consonants are short duration phonemes that are usually misclassified. In order to reduce the error rate and the phoneme confusion, some additional information, such as the bigram model and the phonemes duration, are added to the recognizer during the recognition process (Liang et al., 2006).

State–of-the-art continuous Arabic speech recognition (ASR) systems recognize modern standard Arabic (MSA) that usually appears in TV-News, newspapers, and books. MSA is the formal style of writing and speech across the Arab world (Alghamdi et al., 2009).

Arabic dialects and accents, however, pose great challenges in building automatic speech recognition systems (ASR). For example, MSA-based ASR systems suffer from high word error rate (WER) when applied to those dialects and accents.

This chapter presents our study to examine automatic data driven-based transcription of the Arabic phonemes by developing a more suitable phoneme recognition process. This is accomplished through investigating the possible use of different number of HMM states, and different number of Gaussian Mixtures, on the accuracy and speed of recognizing each phoneme. We have conducted multiple experiments with different parameters, using the Hidden Markov Toolkit (HTK). We used four parameters in the experiments: the number of HMM states for each phoneme, the feature vector type and length, the existence or absence of a language model, and the number of Gaussian mixtures for each phoneme.

The efficiency and correctness of continuous Arabic speech recognition systems hinge on the accuracy of the language phoneme set. The main goal of the work behind this chapter is to recognize and transcribe Arabic phonemes using a data-driven approach. We used the HTK tool to develop a phoneme recognizer, carrying out several experiments with different parameters, such as varying number of HMM states and Gaussian mixtures to model the Arabic phonemes and find the best configuration. We used the corpus presented in CHAPTER 4. As we have sated there, the corpus consists of about 4000 files, representing 5 recorded hours of modern standard Arabic of TV-News. In order to determine the best number of states necessary to represent each phoneme, we carried out a statistical analysis for the phonemes length, frequency and mode (See CHAPTER 4 for more details). A Phoneme recognition accuracy of 56.79% was reached without using any phoneme bigram model. Whereas, the recognition accuracy increased to 96.3% upon using phoneme bigram model.

The remainder of this chapter is organized as follows: Section 2 describes the problem of Arabic phoneme transcription. Section 3 gives an overview of related work to this research. Section 4 details the research methodology and our experimental setup. Section 5 presents the results of our experimentations. Finally, we state our conclusions in Section 6.

5.2 Arabic Phoneme Transcription Problem

Considerable research effort has been conducted in continuous Arabic speech recognition using HMM with varying rates of accuracies. The recognition accuracy in our case is measured by the percent of correctly recognized phonemes. The ASR accuracy is usually affected by several factors, including the phoneme set used, the number of HMM states for each phoneme and number of Gaussian mixtures. Using efficient Arabic phonemes transcription in continuous Arabic speech recognition before using the language model increases the recognition speed and reduces the memory consumption caused by the recognizer. In addition, determining the number of HMM states to model each phoneme increases the accuracy of the recognition. It may also reduce the HMM chain for each utterance. The reason is that some phonemes need only one HMM emitting state to be modeled, while others may need two or more emitting states.

In this chapter we propose to build a phoneme recognizer based on a data driven approach using the HTK tool. Different number of Gaussian mixtures and different number of HMM states are investigated in order to reach the best configuration model for each phoneme.

5.3 Arabic Phoneme Transcription in the Literature

Phoneme recognition and transcription in continuous speech recognition have been addressed by many researchers. (Lee and Hon, 1989) used a large network of trained phonemic HMMs where the maximum likelihood state sequence of the network gave the recognized phoneme sequence. A multi-state fixed structure with three observation probability distribution represented a phoneme. The acoustic variability was better characterized, even though the phonemes duration and structure were not modeled accurately. (Sitaram and Sreenivas, 1994) used a single large continuous variable duration hidden Markov model (CVDHMM) with a number of states equal to the number of phonemes in the vocabulary. A single CVDHMM state characterizes the whole phoneme. The state models both, the acoustic variability and the durational variability of each phoneme. The researchers had reported a reasonable performance because of the external knowledge incorporated into the model, such as the phoneme duration and the bigram probabilities. (Elmahdy, M. et al., 2011) proposed an approach for rapid phonetic transcription of dialectal Arabic, where the Arabic Chat Alphabet (ACA) was used. The proposed approach represented an alternative way to conventional phonetic transcription methods. Results showed that the ACA-based approach outperforms the graphemic baseline while it performs as accurate as the phoneme recognition-based baseline with a slight increase in WER. (Bayeh et al., 2004) presented a study of manual and data driven association of words and phonemes. Their study was based on defining the speech recognition system in a target language based on acoustic models from another source language. They found that phoneme-to-phoneme association was more practical but, words transcription provided better results. Experiments were conducted with French as the source language and Arabic as the target language.

(Alghamdi et al., 2009) proposed a new labeling scheme which was able to cover all the Quranic Sounds and their phonological variations. They presented a set of labels that covers all Arabic phonemes and their allophones. They showed how it can be used efficiently to segment the Quranic corpus. The authors claimed that the initial results were encouraging.

5.4 Methodology and Experimental Setup

The methodology we followed to transcribe Arabic phonemes in a data-driven approach is shown in Figure 5-1. The methodology involves the following steps:

1. Data preparation, which includes:
    a) Defining a proper finite-state recognition grammar.
    b) Building the HMM models (Prototypes) for the Phonemes.
2. Training the HMM models.
3. Phoneme recognition and transcription.
4. Experimentation to find the best HMM configuration.
5. Adding the language model.

Details of Tasks 1 – 3 are given in the remainder of this section. Section 5 explains Tasks 4 and 5.

5.4.1 Data Preparation

In all followed experiments done in this chapter, we depend on the Arabic corpus described in CHAPTER 4 – Section 2.

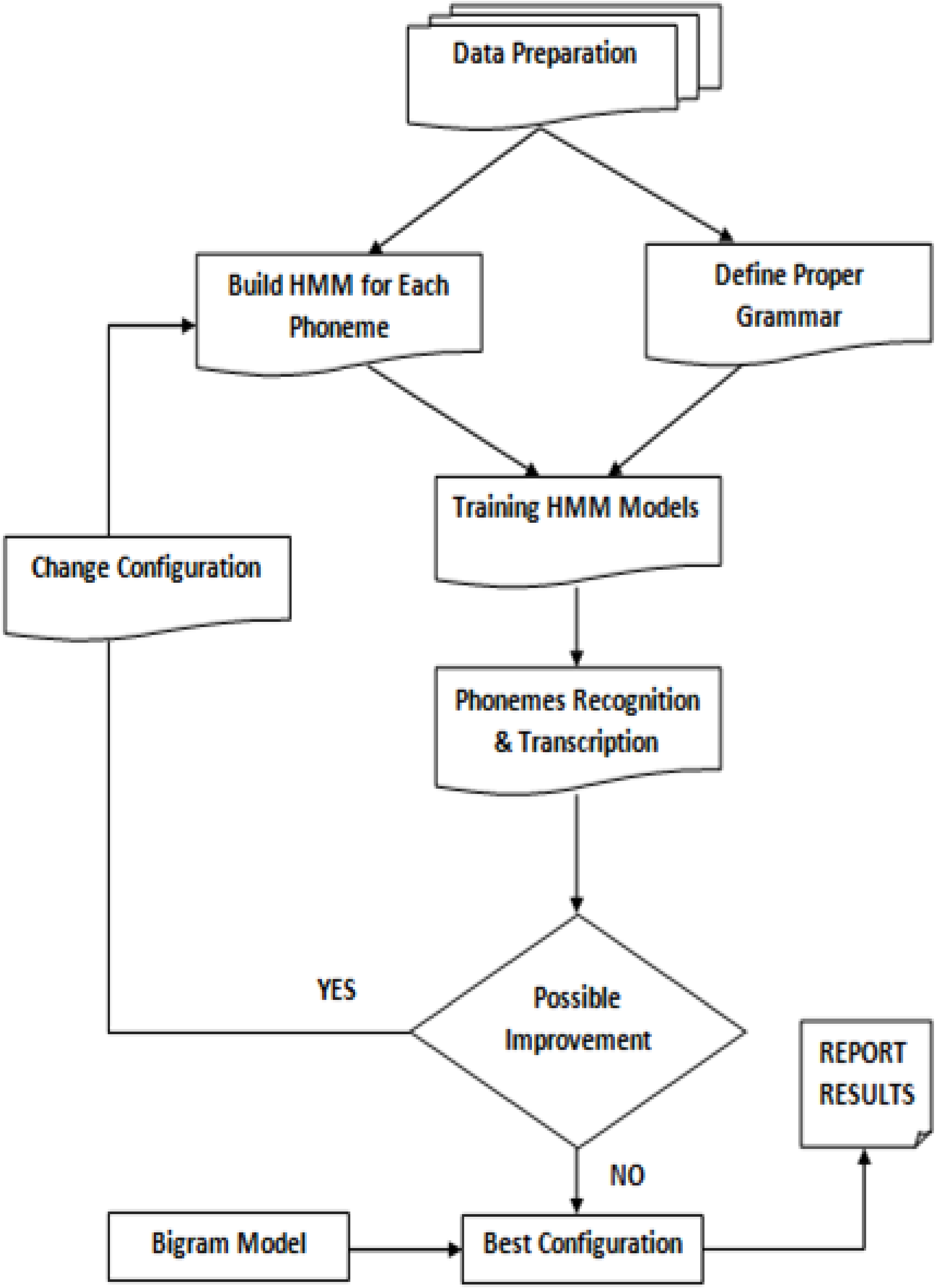


Figure 5-1: Arabic Phoneme Transcription Methodology

Using CMU sphinx, the trigram word transcription and time alignment of all utterances of the corpus were produced. Table 5-1 shows the transcription of the word "ارتفعت" (**E IH R T AE F AE AI AE T**) as an example of trigram word transcription generated using CMU sphinx. Providing the CMU Sphinx with the MFCC feature files of the training set corpus and their corresponding transcriptions, a mechanism of forced alignment was done by Sphinx to generate the phoneme and 'word' segmentation for all utterances of the training set. The testing set of utterances used in phoneme transcription was not seen before for the purpose of evaluation.

CMU Sphinx produced a start and end frame numbers that are associated with each phoneme utterance, as shown in Table 5-1.

Table 5-1: The word "ارتفعت" on the (Left) part, trigram word phoneme segmentation on (Right) part.

CMU Sphinx Labeling							
Word Segmentation			Phoneme Segmentation				
Start Frame	End Frame	Word	Start Frame	End Frame	Phone	Previous	Next
0	12	<s>	0	12	**SIL**		
13	15	<s>	13	15	**SIL**		
16-59		ارَتَفَعَت	16	21	**E**	SIL	IH
			22	24	**IH**	E	R
			25	27	**R**	IH	T
			28	30	**T**	R	AE
			31	33	**AE**	T	F
			34	36	**F**	AE	AE
			37	39	**AE**	F	AI
			40	44	**AI**	AE	AE
			45	51	**AE**	AI	T
			52	59	**T**	AE	E (depending on next phoneme)

The HTK, on the other hand, uses a start and end frame numbers based on integer multiples of 100ns, as shown in Table 5-2. Since the output of the CMU Sphinx is fed as input to the HTK tool, we had to transform the CMU sphinx frame labeling into its equivalent HTK tool labeling and remove all other columns. Table 5-2, shows the HTK tool labeling of the word “ارتفعت”.

Table 5-2: The corresponding HTK labeling of the word (ارتفعت) of Table 5-1.

HTK Labeling Based on 100ns Labeling		
Start Frame	End Frame	Phone
0000000	1200000	**SIL**
1300000	1500000	**SIL**
1600000	2100000	**E**
2200000	2400000	**IH**
2500000	2700000	**R**
2800000	3000000	**T**
3100000	3300000	**AE**
3400000	3600000	**F**
3700000	3900000	**AE**
4000000	4400000	**AI**
4500000	5100000	**AE**
5200000	5900000	**T**

Another issue to be handled was the incompatibility of the MFCC feature files included in the corpus and the MFCC feature files that were accepted by the HTK tool. Therefore, we had to regenerate the MFCC files from the WAV files of the corpus using the HCopy command in HTK (Young et al., 2006). It is worth mentioning that the HCopy command requires proper configuration parameters according to the input WAV files. Table 5-3 shows the configuration parameters that were most suitable to the corpus WAV files.

Table 5-3: Chosen configuration Parameter values for HTK-HCopy Command

HTK Tool - HCopy Configuration Parameters	
Variable	Value
TARGETKIND	MFCC_0_D_A
TARGETRATE	100000.0
SAVECOMPRESSED	F
SAVEWITHCRC	F
WINDOWSIZE	250000.0
USEHAMMING	F
PREEMCOEF	0.97
NUMCHANS	26
CEPLIFTER	22
NUMCEPS	12
ENORMALIZE	F

HTK supports many different types of MFCC feature vectors. The two most-commonly used types in speech recognition are the basic type, MFCC_0 with 13 features frame length, and the extended one, MFCC_0_D_A with 39 features frame length (Huang et al., 2001). Since CMU Sphinx uses the extended one, we selected the MFCC_0_D_A type[1].

Finally, the corpus was divided into two parts; one part for training and the other part for testing. The size of training part was 70% of the corpus. The remaining 30% of the corpus was used for testing.

5.4.2 Building HMM Models

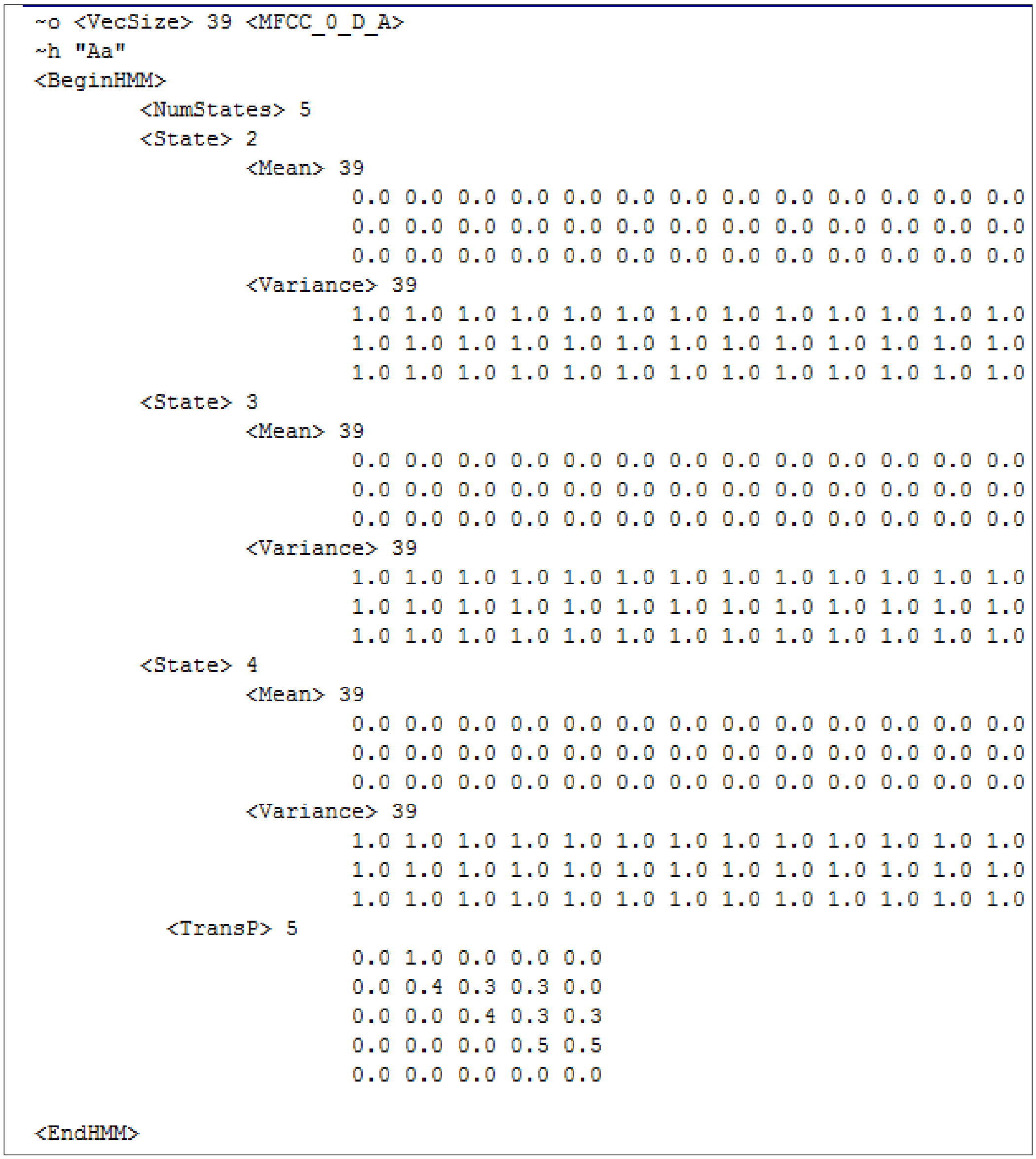
We created an HMM definition file (Prototype) for each Arabic phoneme. Initially, we used 3-emitting state HMM with one Gaussian Mixture for all the phonemes. The 3-emitting state HMM is represented by 5-states in the HTK tool (one entry state, 3 emitting states and one final state). The feature vector used was MFCC_0 with 13 feature values for each frame. Figure 5-2 shows a simple HMM prototype file of 4 emitting states, and one Gaussian mixture. The function of the prototype is to describe the form and topology of the HMM. The initial numbers representing transition sate probabilities used in the definition are not important since they will be updated by the training process.

Different numbers of Gaussian mixtures were used along with different numbers of HMM states in modeling Arabic phonemes. We will first evaluate the accuracy without using any phoneme bigram model. Then, we will add the language model to the configuration of the highest accuracy.

---

[1] The ‘0’ character means 13 features per frame, ‘D’ character stands for Delta (first difference) and A stands for Acceleration (Second difference).

In Figure 5-2, the HMM name is “Aa” which represents the phoneme or sub-word unit. The transition probability matrix (TransP) which represents the last non-emitting state in the HMM definition is important to understand and imagine the HMM states behavior. In Figure 5-2 TransP says that the HMM model starts at state 2 with initial probability 1.0 (sure probability) and may stay in the same state with probability of 0.4 or transit to second state in 0.3 probability and may skip second state and go to third state in probability of 0.3



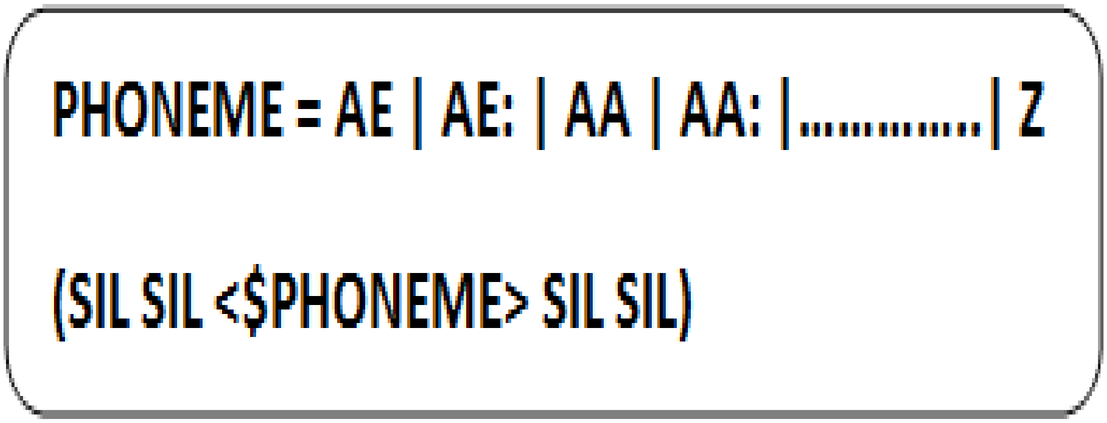
```
~o <VecSize> 39 <MFCC_0_D_A>
~h "Aa"
<BeginHMM>
        <NumStates> 5
        <State> 2
                <Mean> 39
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                <Variance> 39
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
        <State> 3
                <Mean> 39
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                <Variance> 39
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
        <State> 4
                <Mean> 39
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                <Variance> 39
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
          <TransP> 5
                        0.0 1.0 0.0 0.0 0.0
                        0.0 0.4 0.3 0.3 0.0
                        0.0 0.0 0.4 0.3 0.3
                        0.0 0.0 0.0 0.5 0.5
                        0.0 0.0 0.0 0.0 0.0

<EndHMM>
```

Figure 5-2: Simple HMM Prototype of 3 Emitting States

5.4.3 Defining proper grammar, dictionary, and Network Lattices

Using HTK grammar definition language, we define the grammar for each Arabic utterance as shown in Figure 5-3.

The first part of the grammar is the variable PHONEME which defines all possible Arabic phones including silence (SIL) and inhalation (+INH+). The character ‘|’ represents the logical OR operator. The second part represents the utterance where two silences are added, at the beginning and at the end of each utterance. At the middle is the reference for the variable PHONEME inside the characters ‘< >’ which represents the repetition. Figure 5-4 shows a mapping between the utterance “ ارتفعت الأسهم الكويتية ارتفاعا طفيفا اليوم السبت” and the defined grammar in Figure 5-3 on the phoneme level. The reading order in Figure 5-4 is assumed from left to write for both the phoneme list and the words list. Of course, each Arabic word is read from right to left.



```
PHONEME = AE | AE: | AA | AA: |………….| Z

(SIL SIL <$PHONEME> SIL SIL)
```

Figure 5-3: Grammar Used in Phonemes Transcription

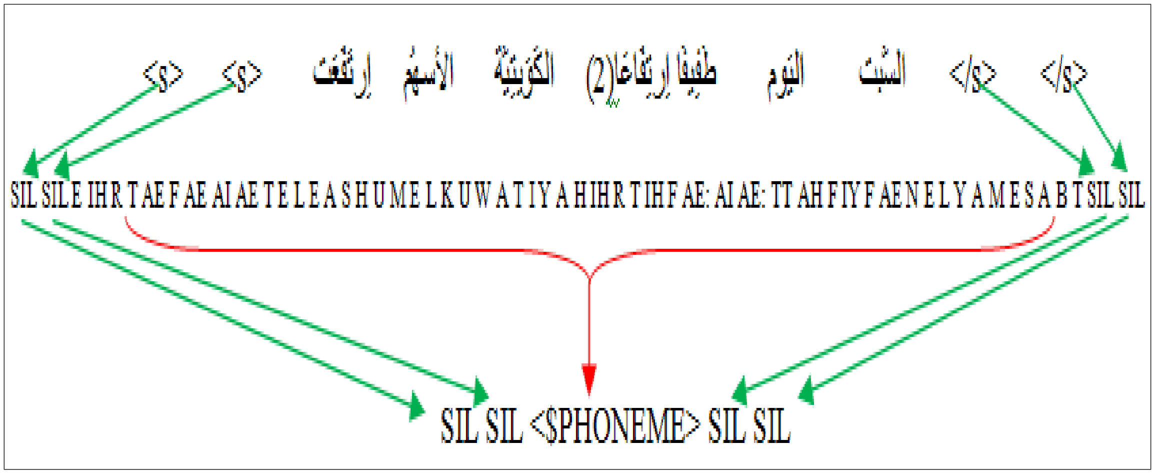
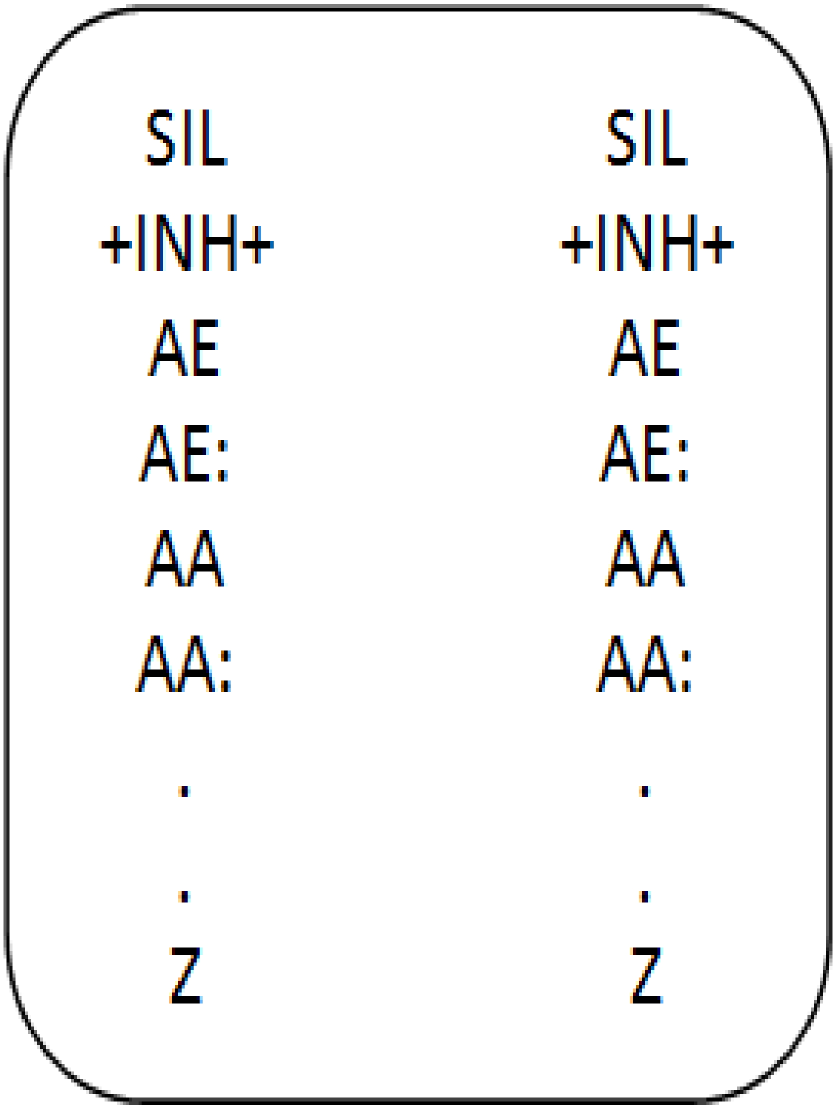

<s> <s> ارتفعت الأسهم الكويتية ارتفاعا(2) طفيفا اليوم السبت </s> </s>
SIL SIL E IHR T AE F AE AI AE T E L E A S H U M E L K U W A T IY A H IHR T IH F AE: AI AE: T T AH F IY F AE N E L Y A M E S A B T SIL SIL
SIL SIL <$PHONEME> SIL SIL

Figure 5-4: Utterance and Grammar Mapping

Using the HTK dictionary definition rules, we define the phoneme dictionary. It consists of two identical columns containing the list of Arabic phonemes based on the HTK tool instructions for recognition at the phoneme level (Young et al., 2006). Figure 5-5 shows part of the dictionary.

The dictionary could be generated automatically through the HTK. It could also be prepared manually using any text editor (Young et al., 2006). The phoneme lattice network is generated after setting the proper grammar and the dictionary. The Lattice network is generated by providing the dictionary and the grammar as input to the HTK-HParse and the HTK-HGen commands. HTK-HGen command is then used to confirm the correctness of the grammar by automatically generate a set of predefined number of utterances.



SIL	SIL
+INH+	+INH+
AE	AE
AE:	AE:
AA	AA
AA:	AA:
.	.
.	.
Z	Z

Figure 5-5: The Dictionary

5.4.4 Training HMM’s

At this point, and after the configuration was set and the grammar was defined, one can start the HMM training process. The HMM models of the phonemes are first initialized using the Baum Welch algorithm. The HTK tool supports two types of initialization. The first type is applied when the boundaries of phonemes are known. The second one is applied when the boundaries of phonemes are not known, which is called flat initialization. Since the phoneme boundaries of our data are known, the first type of initialization is applied using the HInit command of HTK. Figure 5-6 reproduced from (Moreau, 2002), shows the needed inputs and the resulting outputs of the initialization process. For more details on HInit-HTK command see (Young et al., 2006). After initializing all the HMMs, the first iteration of the iterative Baum Welch algorithm is executed using the HTK–HRest command.

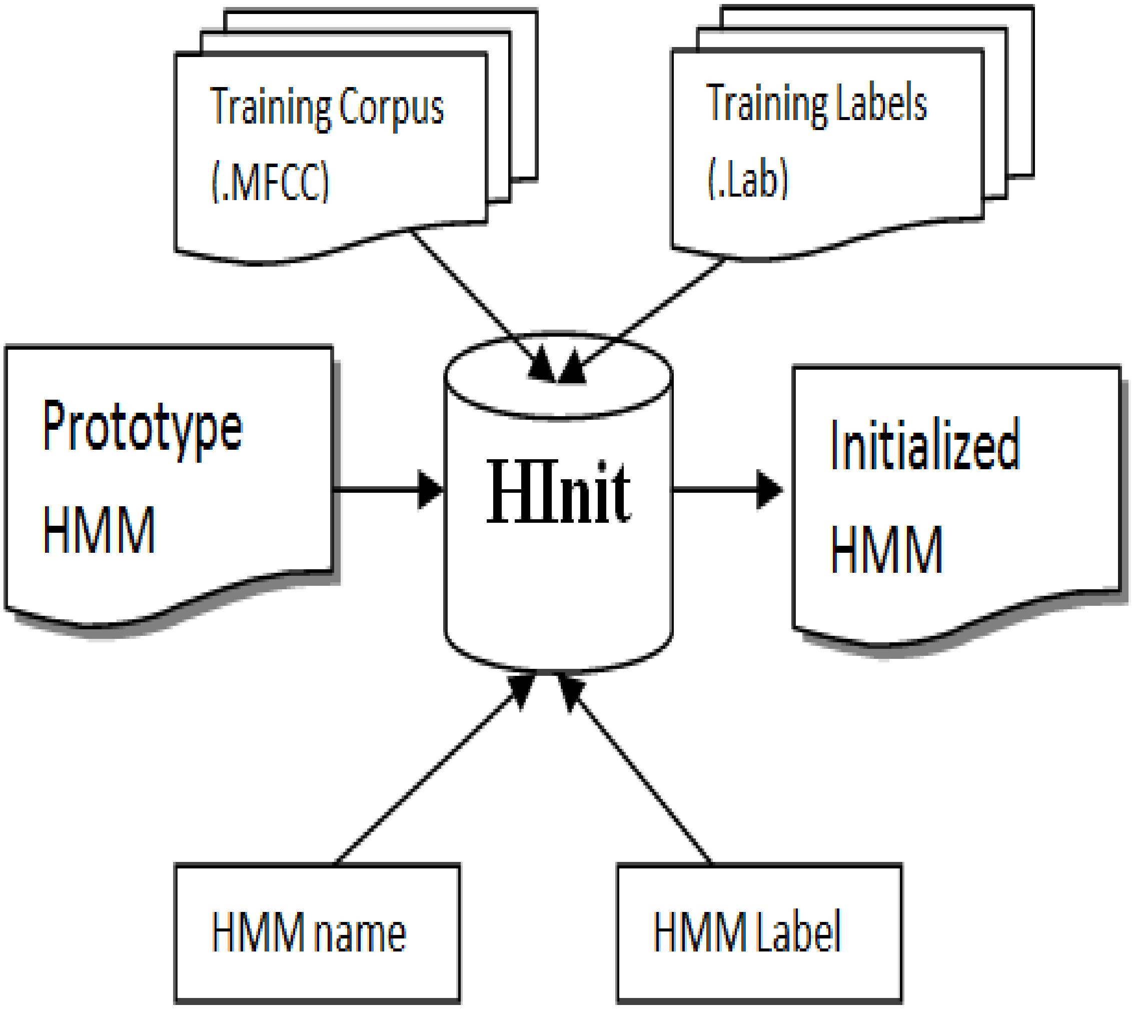


Figure 5-6: Initialization From Prototype (Moreau, 2002).

HTK–HRest command functionality is very similar to the HTK-HInit command, except that it expects that the input HMM definition has already been initialized. It uses the Baum-Welch re-estimation algorithm in place of the Viterbi training (Young et al., 2006). All initialized HMMs are put in one file called the Master Macro File (MMF file). Next, we carry out our phoneme recognition and transcription task shown in Figure 5-1, without using any language model in order to determine the most accurate configuration achievable. Figure 5-7 shows this process details. The training module receives the training MFCC files containing the features of the segmented speech, its corresponding text, and the initialized HMM for each phoneme. The training module produces trained HMMs. The trained HMMs and the defined grammar are used to recognize the MFCC files extracted from the testing set utterances, producing the resulting transcription.

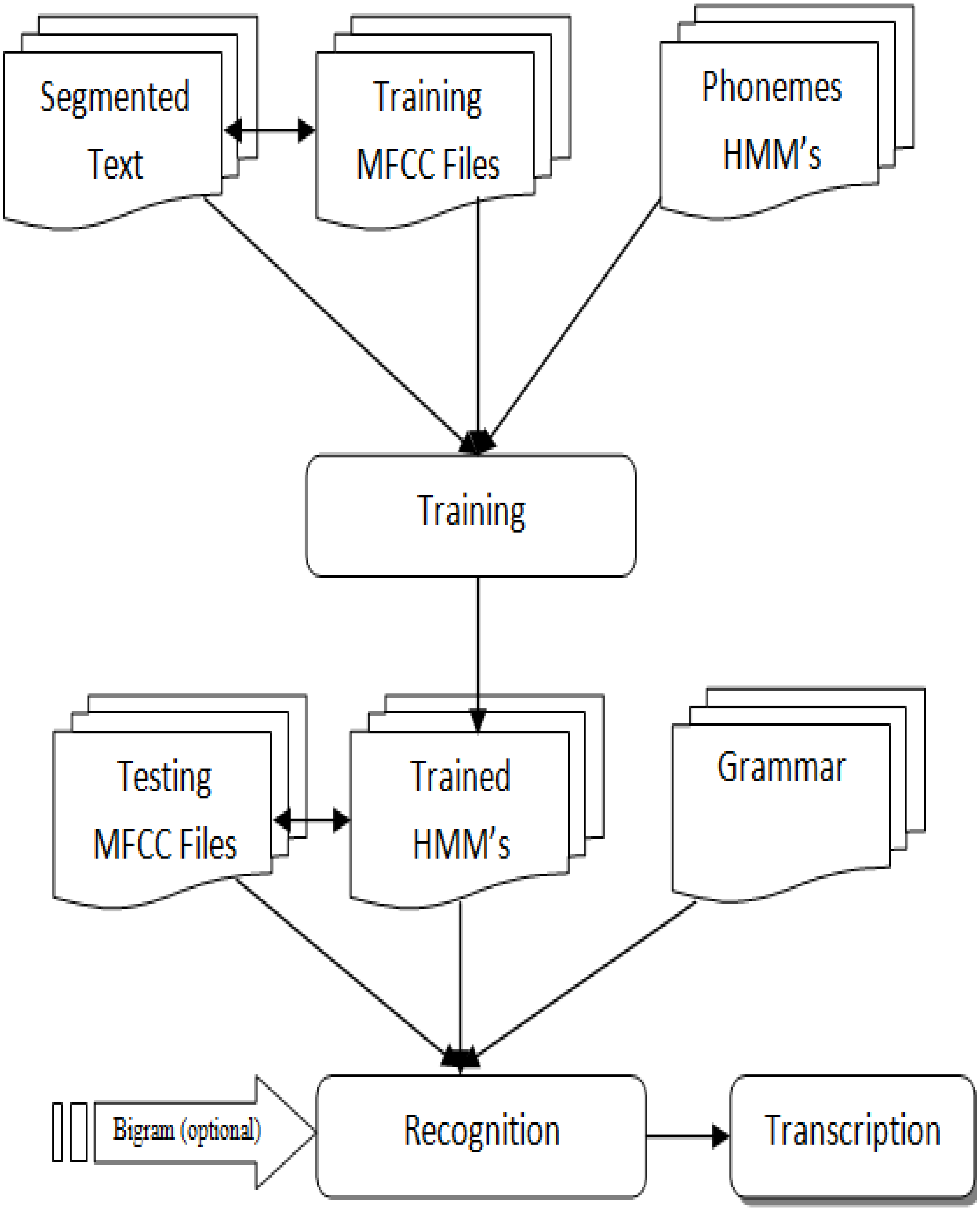


Figure 5-7: Phonemes Recognition and Transcription Process

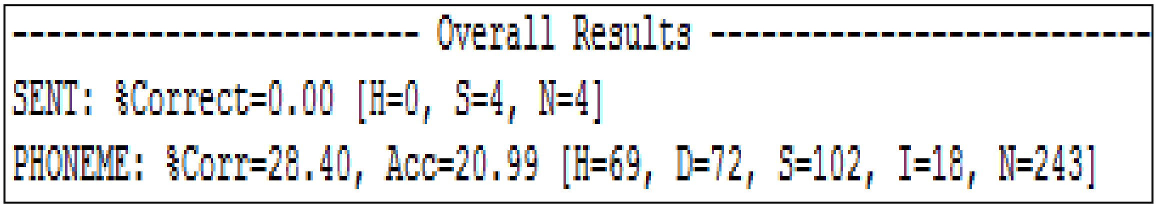
This process is repeated for different HMM configurations until no significant improvement is achieved. At this point, and when needed, the HTK-HBuild command, used to build the language model, adds the bigram probability to the Phonemes network. Details of these experimentations are shown in Section 5.5.

5.5 Experimental Results

A set of experiments were conducted. Each experiment has its own configuration based on the study and analysis of the results of the preceding experiments. A total of 25 experiments were conducted in order to find the best configuration. Below, we report the results of four experiments with highest significant improvements based on our sensitivity analysis. The selection was done manually by investigating the confusion matrix of each experiment.

5.5.1 Experiment 1: Basic Feature Vector with 3 HMM-Emitting States

In this experiment, we used 3 HMM-emitting states for all phonemes, and extracted the 13-value basic features from the WAV files. No Gaussian mixtures were used. The percentage of correctly recognized phonemes was 28.4%. A portion of the output file generated by the HTK-HResult command for aligning the labeled tested utterances with the reference labeling is shown in Figure 5-8. This low recognition rate was due to the usage of the basic features which were not adequate to represent wav files for different phonemes.



```
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=4, N=4]
PHONEME: %Corr=28.40, Acc=20.99 [H=69, D=72, S=102, I=18, N=243]
```

Figure 5-8: Results from First Experiment

Once the HTK-HResult is executed, then, insertion errors (I), substitution errors (S), deletion errors (D) and the total number of labels in the reference transcription (N) are calculated and reported as shown in Figure 5-8.

The H-value in this output refers to the number of correct hits, i.e., the number of correctly recognized phonemes (N-D-S). The percentages of correctness (denoted by %Corr) and accuracy (denoted by %Acc) are based on the values of I, S, D and N. Those percentages are computed based on the following equations (Young et al., 2006):

$$\mathbf{Percent\ Correct} = \frac{\mathbf{H}}{\mathbf{N}} \times \mathbf{100}\% \tag{5-1}$$

$$\mathbf{Percent\ Accuracy} = \frac{\mathbf{H-I}}{\mathbf{N}} \times \mathbf{100}\% \tag{5-2}$$

5.5.2 Experiment 2: Extended Feature Vector with 3 HMM-Emitting States

In the second experiment, we used the extended feature vector instead of the basic one with 4-Gussaian mixtures. The recognition rate increased to 43.3%. We conclude that this significant increase in accuracy is attributed to our use of the extended feature vector of 39 features to represent the wave files along with the use of the 4-Gussaians. Therefore, all subsequent experiments were conducted on the extended feature vector.

5.5.3 Experiment 3: Extended Feature Vector with Varying Number of HMM- States I

In the third experiment, we used the extended feature vector with 4-Gussian mixture. In addition, we used different number of emitting states as follows: The phonemes representing long and intermediate vowels such as /AA:/ , /AE:/ , /AY/ , /UW/ , /AW/ and /AH:/ were assigned 4-HMM emitting states. The un-voiced stop phonemes /E/, /T/, /K/, and /Q/ were assigned 1-HMM emitting state, whereas the voiced-stop phonemes /B/, /D/, and /DD/ were assigned 2-HMM emitting states. The remaining phonemes were assigned 3-HMM emitting states. Table 4-1 shows the complete list of the Arabic phonemes with their corresponding Arabic letters including some examples of using Arabic vowels allophones. The percentage of correctly recognized phonemes in this experiment increased to 51.3%. This shows that using different number of HMM emitting states has a positive impact on recognition accuracy.

5.5.4 Experiment 4: Extended feature vector with varying number of HMM-states II

Based on the confusion matrix of Experiment 3, it was evident that the phoneme /T/ requires no more than one emitting state. In addition, the number of emitting states for Phonemes /E/, /K/ and /Q/ were changed to 3, instead of 1. Similarly, Phonemes /B/, /D/, and /DD/ were changed to 3, instead of 2 emitting states. The number for Phoneme /AA:/ was reduced from 4 to 3 emitting states. Finally, Phonemes /UW/ and /AE/ were assigned 4-HMM emitting states. By applying these changes, the percentage of correctness increased to 56.79 %.

The results of Experiments 3 and 4 highlighted the importance of carrying out a detailed analysis of the various forms a certain phoneme occurs in, in the Arabic speech corpus, in order to systematically determine the most suitable number of HMM emitting states for each phoneme. The next section summarizes our findings with respect to this issue.

5.5.5 Statistical Analysis and Clustering

As we have seen in Chapter 4, a statistical investigation of various phoneme features was carried out. Based on these statistics,

Figure 5-9 shows and example for the shifted distribution of the length probability between long vowel phoneme and the short version of it for the phoneme /AH/.

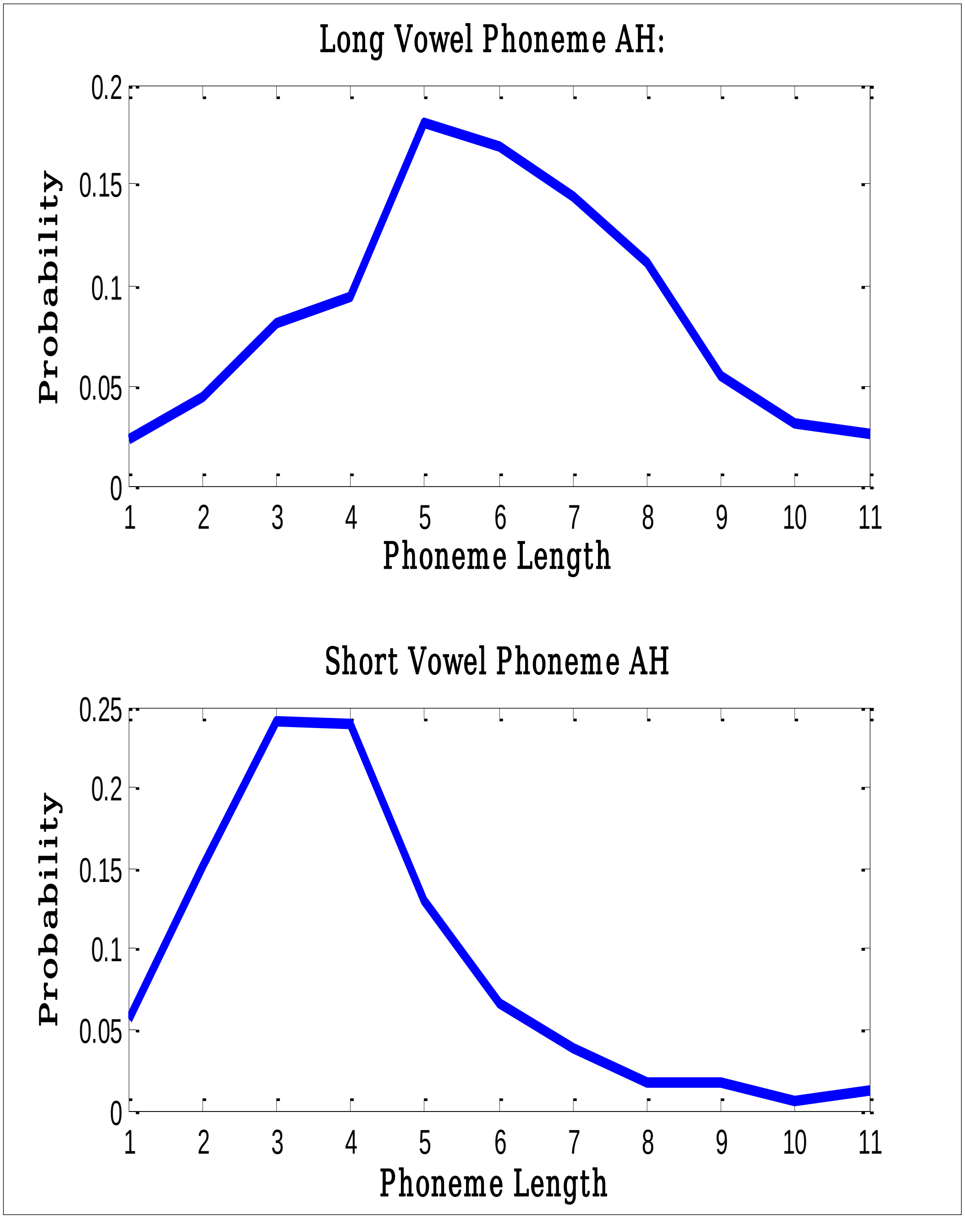


Figure 5-9: Shifted Distribution of the Length Probability Between Short and Long Vowel (AH) Phoneme versions

5.5.6 Experiment with Bigram-Language Model

Based on the results of CHAPTER 4, it is evident that phonemes have been clustered in different statistical features in such a way that one can choose a suitable number of HMM emitting states based on that clustering. It is also clear that using the bigram language model can be helpful in achieving better recognition rates. With respect to the number of HMM states, and based on our experimentations, we chose the phoneme median length results to classify the phonemes based on the assigned number of HMM states.

Each cluster in Figure 4-6 corresponds to number of HMM states. One exception to this is the Phoneme /T/. in which one HMM emitting state is assigned to it based on our previous experimental work. Table 5-4: Arabic Phonemes Clusters shows the various clusters of phonemes and the assigned number of HMM emitting states for each one of them.

Table 5-4: Arabic Phonemes Clusters

Phoneme	Arabic Letter	Number of HMM states	Phoneme	Arabic Letter	Number of HMM
Class 1			**Class 7**		
/E/	ء	2 HMM	/AW/	ـَو	6 HMM
Class 2			/AY/	ـَي	
/IH/	ـِ	3 HMM	/HH/	ح	
/AE/	ـَ		/KH/	خ	
/AA/	ـَ		/S/	س	
/UH/	ـُ		/SH/	ش	
/IX/	ـِ		/SS/	ص	
/R/	ر		/AW/	ـَو	
/L/	ل		/AY/	ـَي	
/N/	ن		/HH/	ح	
Class 3					
/AH/	ـَ	4 HMM			
/B/	ب				
/D/	د				
/DH/	ذ				
/DD/	ض				
/DH2/	ظ				
/M/	م				
/H/	هـ				
Class 4					
/IY/	ـِي	4 HMM			
/UW/	ـُو				
/T/	ت	1 HMM			
/TT/	ط	4 HMM			
/GH/	غ				
/W/	و				
/Y/	ي				
/AE:/	ـَا				
Class 5					
/AA:/	ـَ	5 HMM			
/AH:/	ـَ				
/UX/	ـُ				
/AI/	ع				

/F/	**ف**				
Class 6					
/TH/	**ث**	**5 HMM**			
/JH/	**ج**				
/Z/	**ز**				
/Q/	**ق**				
/K/	**ك**				

In our experimentation, we used 8-Gussian Mixtures except for the /SIL/ model, in which 4-Gussian Mixtures was used. This experiment achieved a phoneme correctness percentage of 96.3%, which is a significant improvement over all our previous experiments that did not include the language model.

5.5.7 Experiments Summary

Table 5-5 summarizes the results of the experiments that have been conducted. Four experiments with variation in the MFCC types, HMM states, Number of Gaussian Mixture and with/ without LM.

Table 5-5: Experiments Summary.

Experiment Number	HMM States	Feature Vector	Vector Length	Language Model	Number of Gaussian Mixtures	Accuracy
1	3	MFCC-0	13	No	1-GM	28.4%
2	3	MFCC-0-D-A	39	No	4-GM	43.3%
3	Different (1,2,3,4)	MFCC-0-D-A	39	No	4-GM	51.3%
4	Changed Based On Exp. 3 with the statistical study	MFCC-0-D-A	39	No	4-GM	56.79%
5	Changed Based On the Statistical Study	MFCC-0-D-A	39	Bigram	4-GM for /SIL/ 8-GM for others	96.3%

5.6 Chapter Summary

In this chapter, we used a data driven approach to recognize and transcribe the Arabic phonemes. We presented the process of phoneme recognition using HMM, implemented with the HTK toolkit. Five experiments were conducted using different HMM parameters. These parameters were mainly set after classifying the phonemes into clusters based on our statistical study for the phonemes length, frequency, median, mode, etc. The maximum phoneme recognition accuracy achieved was 96.3% after employing the bigram language model. A grammar, Phoneme network lattices and a dictionary were generated for the purpose of Arabic phoneme transcription.

The maximum number of Gaussian mixtures used was 8. This number was limited by the size of the existing corpus. It is worth mentioning that more elaborate Arabic speech corpora are still under development with KACST support, which may further improve the recognition rate upon using a higher number of Gaussian mixtures. In addition, we look forward to studying the viability of our proposed methodology on out of vocabulary word recognition, and subsequently on non-MSA (dialect speech) recognition.

Based on the results of this research, and when combining the phoneme recognition with the word language model, we expect a significant enhancement in the continuous Arabic speech recognition.

CHAPTER 6

ARABIC PHONEME RECOGNITION USING LVQ

6.1 INTRODUCTION

In order to improve the automatic speech recognition (ASR) systems for continuous Arabic speech, we have to develop accurate Arabic phoneme recognizer. The core of an accurate Arabic phoneme recognizer is an efficient phoneme classifier. One of the strongest and powerful algorithms for this purpose is the Learning Vector Quantization (LVQ) algorithm that was developed by (Kohonen, 1988).

The LVQ showed excellent performance on the phoneme level recognition in various languages. Also, some Artificial Neural Network (ANN) algorithms such as time delay neural network (TDNN) and Multi-Layer Back Propagation Neural Networks (MLBPNN) showed similar performance in this regard (Kohonen, 1988).

LVQ has some powerful features like; efficiency, speed, and simplicity which made it one of the target solutions for large vocabulary speech recognizers that usually involve large training sets. The recognition rates using LVQ are comparable to those reported by TDNN (Waibel et al., 1989).

LVQ uses feature vectors to split the vector space in a specific classification problem with a set of decision lines that are close enough to the optimal Byes rule. Using LVQ showed high performance in the phoneme classification for even a small number of reference vectors (Kohonen, 1988) and (Yokota et al., 1988).

The main goal of this chapter is to study the viability of using the learning vector quantization (LVQ) classifier to efficiently recognize Arabic phonemes in order to enhance the continuous Arabic speech recognition rate.

The speech recognition system consists of two levels: the acoustic level and the language model. The accuracy of the acoustic model hinges on the accuracy of the phoneme recognizer, which makes it worthy to investigate the use of the LVQ in phoneme recognition.

In this chapter, we propose the use of Learning Vector Quantization (LVQ) algorithm to recognize Arabic phonemes in an open-vocabulary continuous Arabic speech corpus of different TV news clips. We employ a data driven approach to generate the training feature vectors that embed the frame neighboring correlation information. Next, we generate the phoneme codebook using the K-means splitting algorithm. After that, we train the generated codebook using the LVQ algorithm. We use the trained code book in Arabic utterance transcription without adding any phoneme bigrams or learning model. Achievement and results are explained in the rest of this chapter.

The remainder of this chapter is organized as follows: section 6.2, which overviewed the LVQ in the literature. Section 6.3 introduce our methodology. Section 6.4 explains data clustering based on LVQ. Section 6.5 explains the training process of the codebooks. In section 6.6 a performance evaluation was carried out for the phoneme recognition. we started our experimental work at section 6.7. In section 6.8 Arabic utterance transcription using trained codebooks was checked and finally, the chapter summary at section 6.9.

6.2 LVQ in the Literature

The LVQ classification algorithm was used in many applications other than speech recognition. For example, (Prasad and Kohli, 2010) list audio and data compression, facial recognition, radar and signals, finance, and insurance as some of these applications. In addition, they conducted a set of experiments in bioinformatics to group genes using LVQ. They have reported that LVQ algorithm exhibited consistency and better accuracy compared to other clustering techniques like SOM, HC and K-means. They achieved 91.8% accuracy for genome grouping using the LVQ.

With respect to speech recognition, this classifier was applied in ASR for different languages (Kohonen, 1988), (Yokota et al., 1988), (Avdagic et al., 2007), (Essa et al., 2008a) and (AbuZeina et al., 2011). (Yokota et al., 1988) developed a shift-tolerance neural network architecture based on Kohonen's LVQ for phoneme recognition. The LVQ network aimed to recognize the Japanese phonemes in a speaker dependent environment. The recognition performance ranged from 98% to 99%. They showed that the traditional classifiers like K-means classifier and Time Delay Neural Network (TDNN) exhibit less performance as compared to the LVQ.

(Avdagic et al., 2007) presented a new approach for Bosnian phoneme recognition based on a hybrid LVQ/Elman NN. They were able to achieve recognition rates of 99.35%, 98.26%, 97.29%, 98.02% and 99.15% for the five tested Bosnian phoneme classes. These results were reported to be better than the standalone Elman NN-based recognition.

(Kondo et al., 1994) utilized the LVQ algorithm using words instead of phonemes. They applied the recognition process on both isolated and continuously spoken Japanese digits. The performance of the isolated digits was 99.2% and was 95.4% for the continuously spoken digits.

(Mäntysalo et al., 1994) produced a new model for phonemic transcriptions of Finish spoken utterances by combining LVQ and discrete HMM. Using discrete HMM for decoding, they eliminated the need to employ context dependent acoustic information such as; triphones. They used multiple ways for forming high dimensional context vector by both averaging and concatenating short-time feature vectors within time domain windows. Using these high dimensional vectors, trained codebooks were produced by the LVQ. Then, codebooks were combined with the discrete HMM. The phonemic recognition accuracy raised from 87% for short-time feature vectors to 99% when using high dimensional context vectors.

(Ghargen, FS. et al., 1995) addressed the phoneme recognition problem using feed-forward neural network (FWNN). They investigated two types of networks: local neural networks that use linear activation functions in their hidden units; and distributed neural networks, where the hidden units use sigmoid non-linear activation function. They carried out their phoneme recognition experimentation on speaker-dependent Japanese vocabulary using four local neural networks (k-nearest neighbor, LVQ, grow and learn, and Gaussian-based weighted approximation) and one distributed neural network (back propagation). They compared between these methods according to three measurements: classification correctness, network size, and learning time. They found that distributed networks generalize better than local networks but require longer training time, while local networks learn very quickly, but do not generalize well. Among the four local networks, LVQ showed recognition performance of 91.3% compared with the 95.8% for the distributed neural networks.

(Cosi et al., 2000) proposed a phoneme classifier for standard Italian speech using the AIDA standard speech database. They proposed a unified view of Back propagation and LVQ learning schemes. They developed a heuristic learning algorithm referred to as competitive Back propagation for introducing a sort of competition among the locally tuned units of radial basis function (CRBF). Two measures of comparison were used: the first one was the weighted performance which is the total number of correctly classified frames to total number of frames. The second measurement was the un-weighted performance which is the average accuracy over the number of classes. The LVQ Weighted performance for phoneme recognition was 64.2% whereas the Un-weighted performance was 52.4%. The recognition accuracy using a CRBF network trained with competitive back propagation was 67.9% weighted performance, and 53.3% Un-weighted performance. They concluded the effectiveness of the heuristic learning algorithm based on the reported results.

(Kurimo, 1997) conducted some experiments on neural network algorithms including the LVQ to improve phoneme recognition using mixture density hidden Markov Models (MDHMMs). Learning Vector Quantization (LVQ) was used to increase the discrimination between different phoneme models, both during the initialization of the Gaussian codebooks and during the actual MDHMM training. The experiments with LVQ and self organizing maps (SOMs) showed reductions in both the average phoneme recognition error rate and the computational load. They outperformed the maximum likelihood training and the generalized probabilistic descent (GPD) training techniques.

6.2.1 Context Dependent LVQ recognition of the Arabic phonemes

In the aforementioned literature, the LVQ was used to recognize phonemes in conjunction with other models and methods such as HMM and artificial neural networks. In this Section, we focus on context dependent LVQ recognition of the Arabic phonemes.

.Arabic phonemes differ from the previously cited languages in terms of many phonetic features (Ali et al., 2009). Hence, we believe that the use of LVQ in Arabic phoneme recognition deserve to be investigated.

LVQ has been employed by different researchers in Arabic speech recognition. (Essa et al., 2008b) compared between back propagation and LVQ classifiers with different architectures and parameters to recognize a limited number of isolated Arabic words. They reported 96% recognition accuracy for the back propagation, as opposed to 86.6% for the LVQ. It was not clear how one can extend their work to build an open vocabulary classifier using their approach.

(Ma and Zeng, 2012) suggested an improvement to the LVQ algorithm to recognize speaker independent isolated words. Their modified algorithm utilizes the Linde-Buzo-Gary (LBG) algorithm to design the codebooks, and then randomly selects some speakers and their pronounced words in order to optimize them. They compared the recognition of the LBG and LVQ algorithms to their proposed modified LVQ technique on the Arabic digits database. The results showed that the modified LVQ (SLVQ) gave 94.21 % performance which was higher than both LBG and LVQ algorithms that showed 92% and 65% respectively.

(Selouani and Caelen, 1999) combined LVQ and Time Delay Neural Networks (TDNN) to enhance Arabic speech recognition systems (ASRSs) by giving LVQ the ability to overcome problems resulting from the language particularities such as; emphasis, germination and vowel lengthening. The results showed that the proposed version of the LVQ achieved higher recognition rate than the baseline LVQ. They achieved 87% of average accuracy as opposed to 79% accuracy for the baseline LVQ system. After some sensitivity analysis, they concluded that the hybrid system was capable of perceiving relevant phoneme duration changes and emphasis features, unlike the baseline system.

Most of the reported LVQ work for recognition of Arabic speech is either restricted to isolated words or digits (Essa et al., 2008b) and (Ma and Zeng, 2012). In the next sections, we present our study on using standalone LVQ on a large vocabulary continuous Arabic speech corpus. The next section (Section 6.3) presents the methodology used in this study.

6.3 Methodology

The methodology we followed for the Arabic phoneme recognition is based on the LVQ and involves the following phases outlined in Figure 6-1:

1. Data preparation phase.
2. Data clustering phase.
3. Training the codebooks using LVQ algorithm.
4. Performance evaluation of the phoneme recognition.

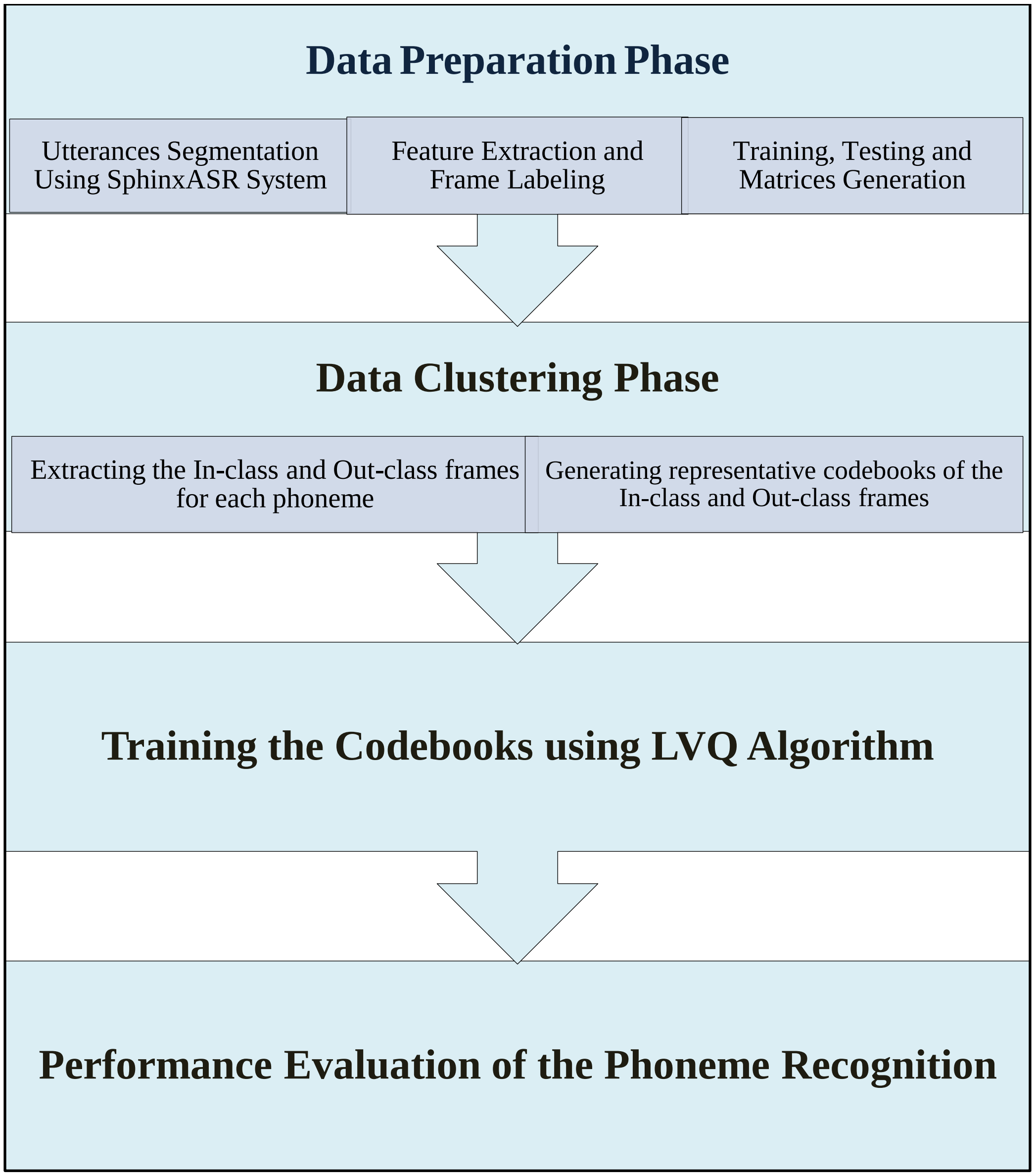

Data Preparation Phase
Utterances Segmentation Using SphinxASR System
Feature Extraction and Frame Labeling
Training, Testing and Matrices Generation
Data Clustering Phase
Extracting the In-class and Out-class frames for each phoneme
Generating representative codebooks of the In-class and Out-class frames
Training the Codebooks using LVQ Algorithm
Performance Evaluation of the Phoneme Recognition

Figure 6-1: Arabic Phoneme recognition Mo Methodology Phase

6.3.1 Data Preparation

In the data preparation phase, we used the corpus described in CHAPTER 4. The data is segmented, and then features are extracted. After that, the data goes through clustering and LVQ training or testing. The following sections describe these processes. The Arabic phoneme set used is the same set described in CHAPTER 4.

6.3.1.1 Utterance Segmentation Using CMU Sphinx ASR System

Providing the CMU Sphinx ASR system see (Lamere et al., 2003) with the MFCC feature files of the training set corpus and their corresponding transcriptions, a mechanism of forced alignment was performed to generate the phoneme segmentation for all utterances of the training set (Al-Manie et al., 2010). We applied the same mechanism using CMU Sphinx on our data. Table 6-1 shows the first 60 frames of the phoneme segmentation of the following utterance:

" ارتفعت الأسهم الكويتية ارتفاعا طفيفا اليوم السبت"

And has the transcription:

“E IH R T AE F AE AI AE T E L E A S H U M E L K U W A T I Y A H IH R T IH F AE: AI AE: TT AH F IY F AE N E L Y A M E S A B T”

Table 6-1: First 60-Frames of the Phoneme Segmentation Using CMU-Sphinx for the Utterance: "ارتفعت الأسهم الكويتية ارتفاعا طفيفا اليوم السبت"

Sphinx Phoneme Segmentation			
Start Frame	End Frame	Segment Score	Triphones
0	12	1125260	SIL
13	15	186438	SIL
16	21	-263197	**E** SIL IH b
22	24	-187945	**IH** E R i
25	27	-113562	**R** IH T i
28	30	-102781	**T** R AE i
31	33	-170048	**AE** T F i
34	36	-218105	**F** AE AE i
37	39	-41958	**AE** F AI i
40	44	153334	**AI** AE AE i
45	51	-28882	**AE** AI T i
52	59	-207607	**T** AE E e
60	65	-207834	**E** T L b

CMU Sphinx produces a start and end frame numbers that are associated with each phoneme utterance, as shown in Table 6-1. Each phoneme is represented using tri-phones, starting with the current phoneme (marked bold in Table 6-1) followed by the previous phoneme and then the next one.

6.3.1.2 Feature Extraction and Frames Labeling

For each utterance we extracted the consecutive frames, in sequence and we stored them in a matrix. After that, we concatenated the matrices of all utterances sequentially in one big matrix, called the features matrix. In parallel with feature matrix generation, we generated another matrix called the frame-label matrix, containing the phoneme label corresponding to each frame (column) in the generated features matrix.

Due to the limited PC memory that MATLAB can use, we were not able to train more than 50000 feature vectors from the feature matrix. The 50000 features were selected so that each phoneme was adequately represented. The distribution of the 50k features was similar to the whole corpus features but with reduced numbers.

Figure 6-2 shows the distribution of all phonemes in the utterances of the training subset. Although we believe that better results would emerge if all training utterances were used, the chosen sample seems appropriate according to the obtained results, shown in Section 6.6.

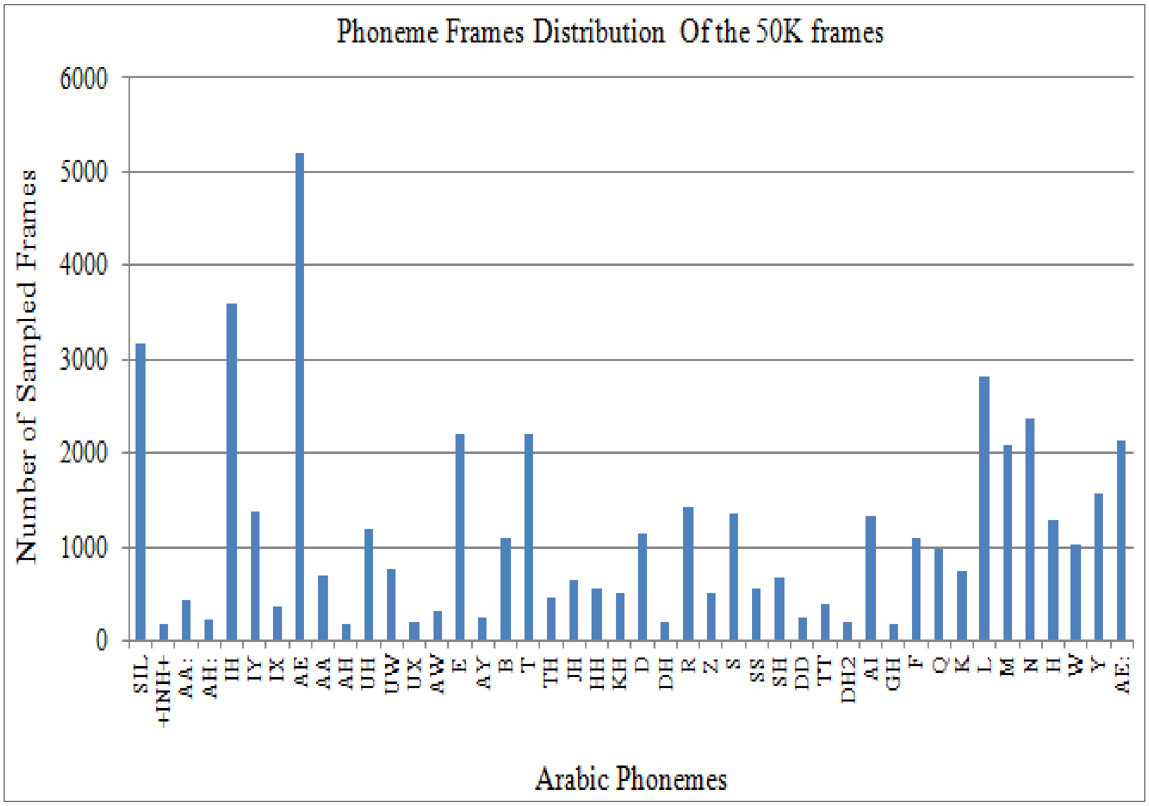


Figure 6-2: Distribution of the phonemes on the chosen utterances from the training set.

With respect to testing, we included all utterances of the testing set in the corpus in batches of 20000 feature vectors, due to memory limitations. In order to implicitly embed the frame neighboring correlation information inside the phoneme feature vector, each frame has been associated with the three preceding frames and the three succeeding frames, generating a feature vector of the current frame and its associated label. Hence, each phoneme-frame[2] in an utterance is represented by a column of 91 features (7X13). The same procedure was applied for all the testing data of the corpus. The complete pictorial view of this process is given by Figure 6-3 where the upper part represents the feature vector matrix and the process of picking the 91 frames to represent the current phoneme. A MATLAB code was written to handle this operation.

---

[2] Except for the first 3 frames and the last 3 frames in the feature matrix.

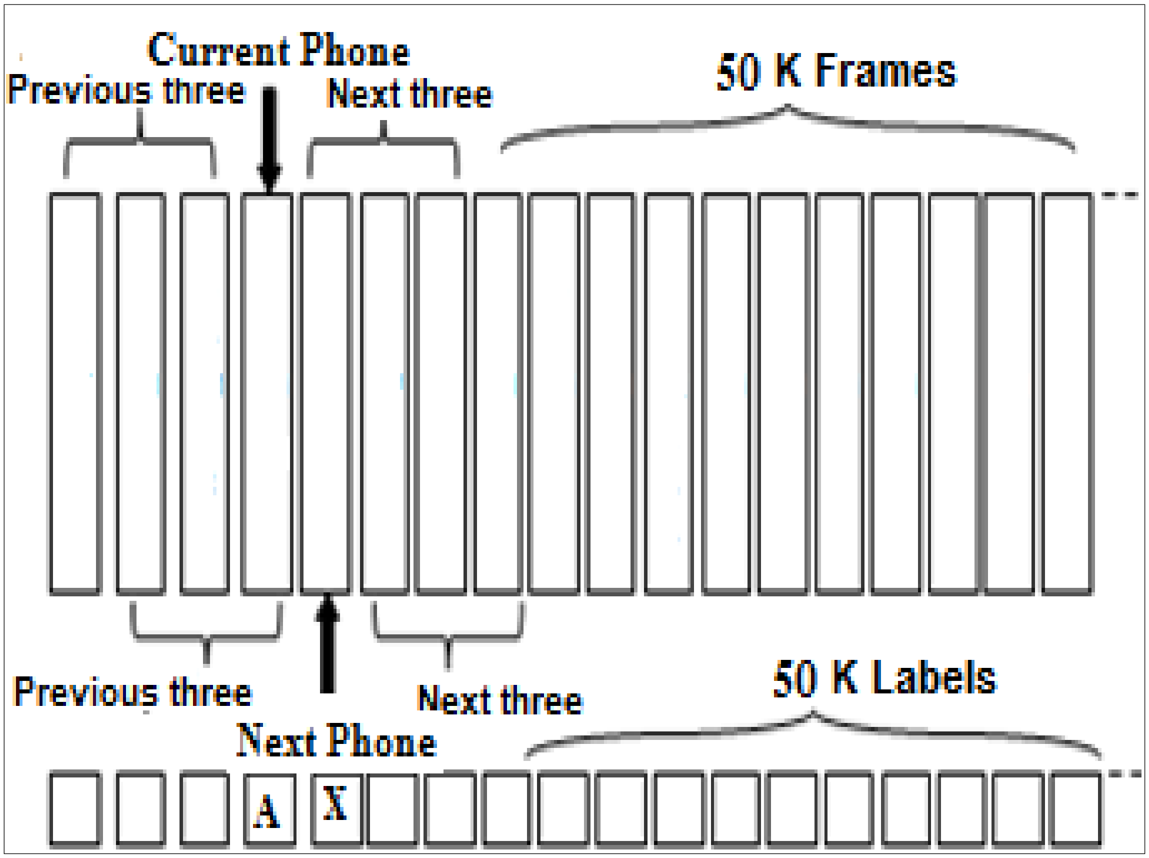


Figure 6-3: Process of training matrix generation

6.4 Data-Clustering

This phase includes two sub-phases; the extraction of the In-Class and Out-Class frames for each phoneme, and the training of the LVQ codebooks.

6.4.1 Extracting In-Class and Out-Class Frames for Each Phoneme

We extracted the In-Class and Out-Class frames for each Arabic phonemes similar to those used in (Ali et al., 2009) from the newly generated feature matrices. Figure 6-4 illustrates the idea. After we have extracting the In-Out frames, we determined the sizes of the codebooks for each type of classes experimentally based on the size of the phoneme frames denoted by (N). Table 6-2 shows our criteria and suggested codebooks sizes. The symbols K_a and K_b stand for the names of the codebooks for the In-class and Out-class respectively. Symbols S_a and S_b are the variables that hold the sizes of the in-class and out-class sizes respectively. The criteria used in the table is based on the size of the phoneme in frames (N), since we only picked 50K sample we divide the sizes based on this set, it depends on the occurrences, usually the out-class size is fixed 128 and the in-class sizes must be power of two.

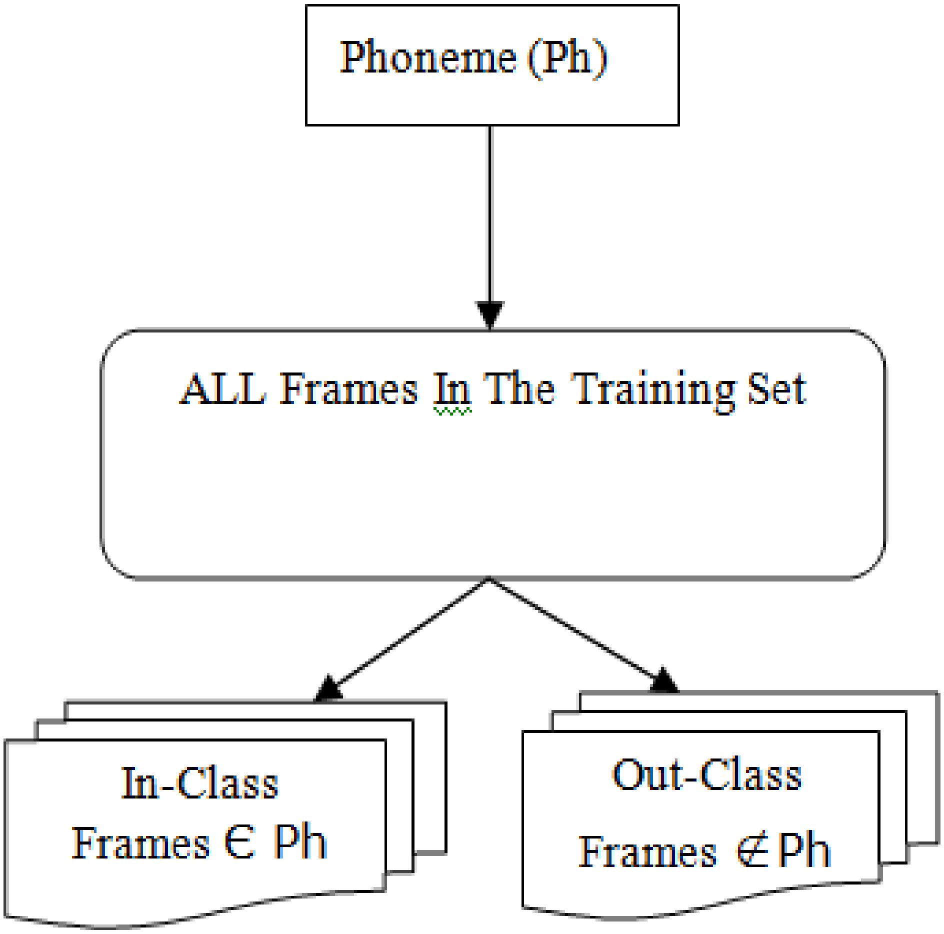


Figure 6-4: Two classes of frames (In-Class & Out-Class)

Table 6-2: Criteria of the code-book sizes

Criteria Based On Phoneme Frames (N)	Size of In-class Codebook (K_a) is S_a	Size of Out-class Codebook (K_b) is S_b
N<16	8	128
16<N<500	16	128
500<N<1000	32	128
1000<N<2000	48	128
Otherwise	64	128

Table 6-2 shows that the values of sizes S_a and S_b were chosen to be power of 2 to improve the efficiency of the computation, and to use it in the splitting K-Mean algorithm that requires binary splitting.

6.4.2 Generating Representative Codebooks

We used the vector quantization K-means algorithm with splitting that was developed by (MacQueen, 1967) to generate the codebooks K_a and K_b of sizes S_a and S_b, respectively.

6.5 Training Code-Books Using the LVQ Algorithm

At this stage, using the Kohonen's Based LVQ algorithm (Kohonen, 1988), we adapted the location of the frames by passing over all frames in the main matrix regardless of whether it belongs to In-Class or Out-Class. Then we calculated the distance between the current frame and the representative codebooks of both In-Class frames and Out-Class frames.

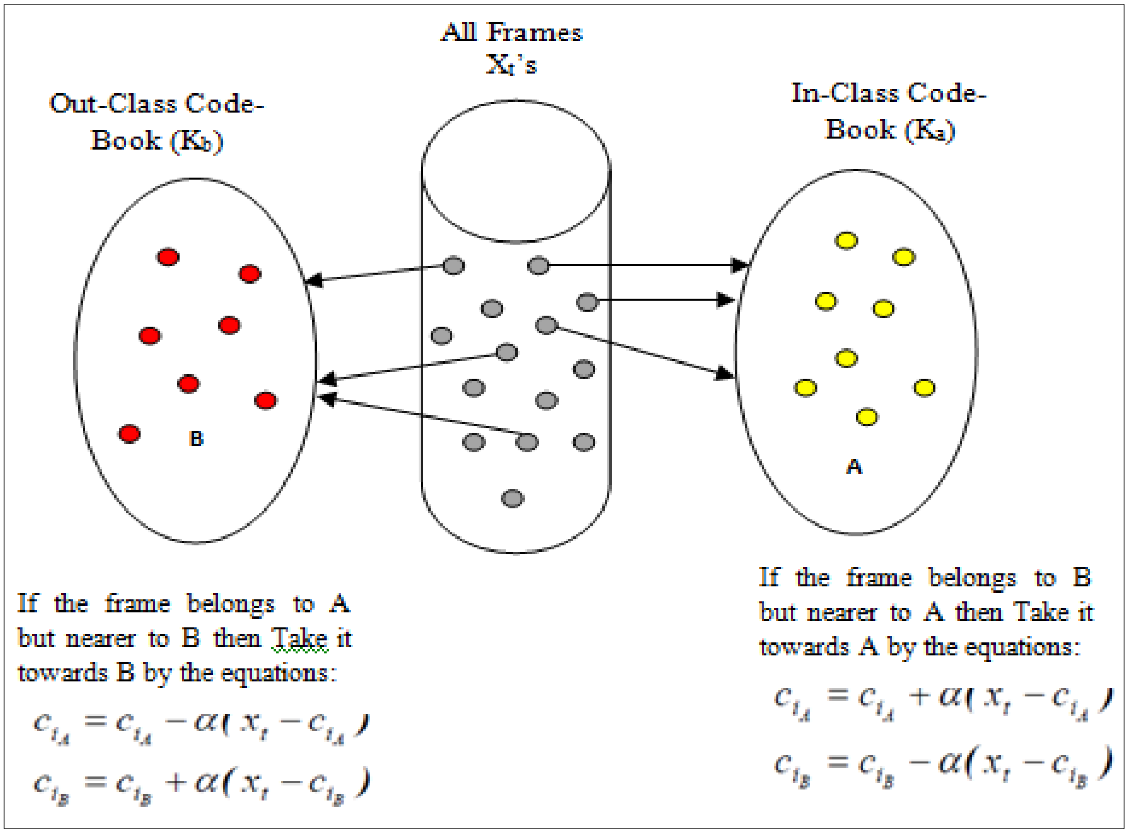
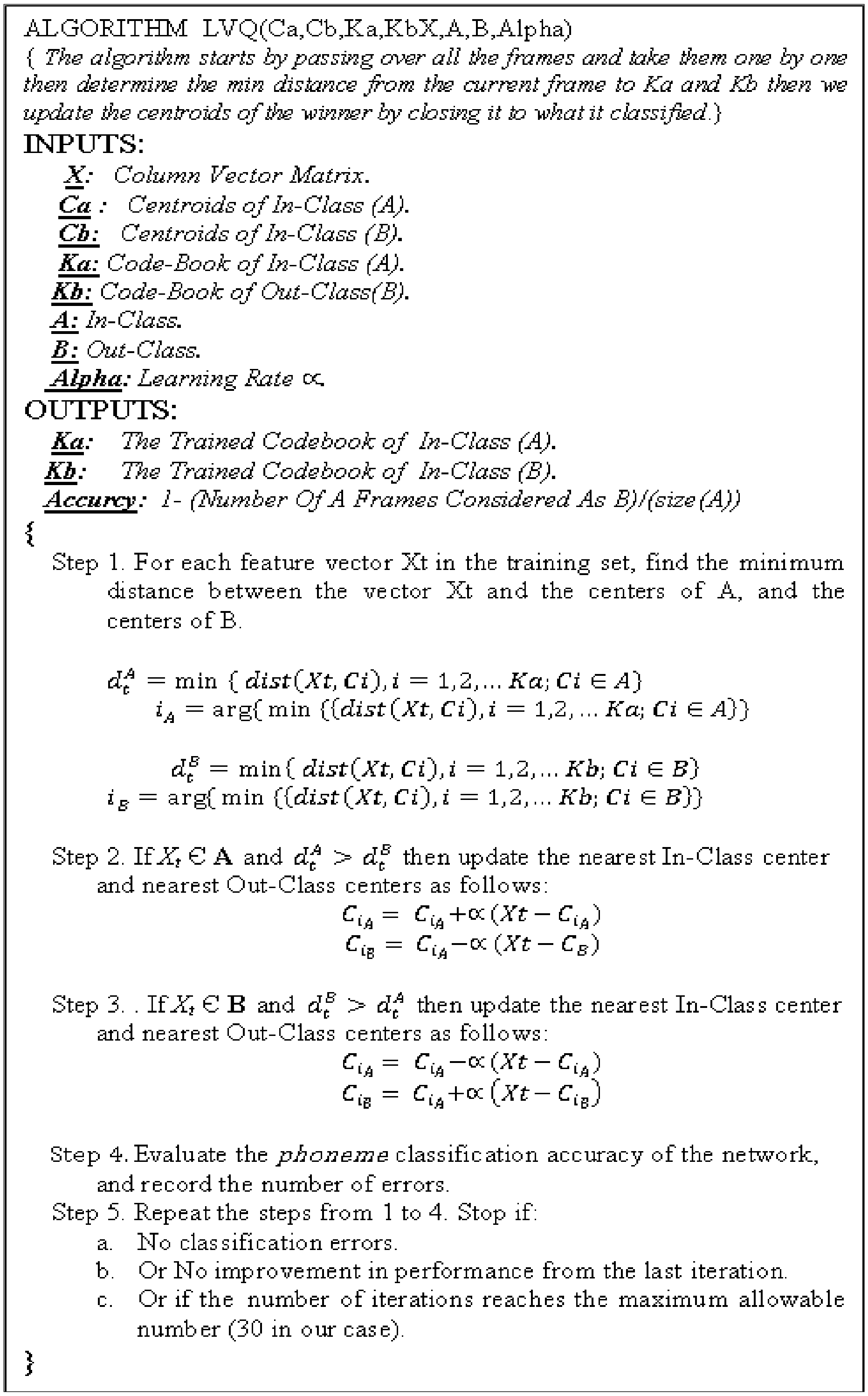

All Frames
X_t's
Out-Class Code-Book (K_b)
In-Class Code-Book (K_a)
B
A
If the frame belongs to A but nearer to B then Take it towards B by the equations:
$c_{t_A} = c_{t_A} - \alpha(x_t - c_{t_A})$
$c_{t_B} = c_{t_B} + \alpha(x_t - c_{t_B})$
If the frame belongs to B but nearer to A then Take it towards A by the equations:
$c_{t_A} = c_{t_A} + \alpha(x_t - c_{t_A})$
$c_{t_B} = c_{t_B} - \alpha(x_t - c_{t_B})$

Figure 6-5: Pictorial View of LVQ Adaptation Step

After that, we applied the adaptation formulas (see Figure 5-6 and Figure 6-6) to push the frame either to be in the In-Class frames codebook (K_a) or in Out-Class frames codebook (K_b) based on the nearest distance. The main steps of the LVQ are shown in Figure 6-6. The LVQ algorithm was applied on all the 44 Arabic phonemes and the performance of the recognition was computed for each one.



```
ALGORITHM  LVQ(Ca,Cb,Ka,KbX,A,B,Alpha)
{ The algorithm starts by passing over all the frames and take them one by one then determine the min distance from the current frame to Ka and Kb then we update the centroids of the winner by closing it to what it classified.}
INPUTS:
    X:  Column Vector Matrix.
    Ca :  Centroids of In-Class (A).
    Cb:  Centroids of In-Class (B).
    Ka: Code-Book of In-Class (A).
    Kb: Code-Book of Out-Class(B).
    A: In-Class.
    B: Out-Class.
    Alpha: Learning Rate ∝.
OUTPUTS:
    Ka:  The Trained Codebook of  In-Class (A).
    Kb:  The Trained Codebook of  In-Class (B).
    Accurcy:  1- (Number Of A Frames Considered As B)/(size(A))
{
    Step 1. For each feature vector Xt in the training set, find the minimum
            distance between the vector Xt and the centers of A, and the
            centers of B.

            $d_t^A = \min \{ dist(Xt, Ci), i = 1,2,\ldots Ka; Ci \in A\}$
                $i_A = \arg\{ \min \{(dist(Xt, Ci), i = 1,2,\ldots Ka; Ci \in A)\}$

                $d_t^B = \min\{ dist(Xt, Ci), i = 1,2,\ldots Kb; Ci \in B\}$
            $i_B = \arg\{ \min \{(dist(Xt, Ci), i = 1,2,\ldots Kb; Ci \in B)\}$

    Step 2. If $X_t \in \mathbf{A}$ and $d_t^A > d_t^B$ then update the nearest In-Class center
        and nearest Out-Class centers as follows:
                $C_{i_A} = C_{i_A} + \propto (Xt - C_{i_A})$
                $C_{i_B} = C_{i_A} - \propto (Xt - C_B)$

    Step 3. . If $X_t \in \mathbf{B}$ and $d_t^B > d_t^A$ then update the nearest In-Class center
        and nearest Out-Class centers as follows:
                $C_{i_A} = C_{i_A} - \propto (Xt - C_{i_A})$
                $C_{i_B} = C_{i_A} + \propto \left(Xt - C_{i_B}\right)$

    Step 4. Evaluate the phoneme classification accuracy of the network,
        and record the number of errors.
    Step 5. Repeat the steps from 1 to 4. Stop if:
        a.  No classification errors.
        b.  Or No improvement in performance from the last iteration.
        c.  Or if the  number of iterations reaches the maximum allowable
            number (30 in our case).
}
```

Figure 6-6: LVQ Algorithm

6.6 Performance Evaluation of Phoneme recognition.

In this phase we analyze K-means algorithm with splitting d the performance of the training process by calculating the minimum distance between each frame and the trained codebooks and assigning them to the proper class. Two types of errors were found; the first one appeared when a frame belonged to A-Class classified to belong to B-Class. The second error type was the vice versa situation. We were concerned about the frames of a phoneme that migrate from A-Class to B-Class. We used those frames in calculating the accuracy or performance of the recognition for the phoneme. The accuracy was measured using the following formula:

$$\mathbf{Accuracy} = \mathbf{1} - \frac{\boldsymbol{\mathit{Number\ Of}}\ (\boldsymbol{A_{\smile}\text{ç}lass}) - \boldsymbol{\mathit{Frames\ Considered\ as}}\ (\boldsymbol{B_{\smile}\text{ç}lass}) - \boldsymbol{\mathit{Frames}}}{\boldsymbol{\mathit{Size}}\ (\boldsymbol{A_{\smile}class})} \tag{6-1}$$

The LVQ algorithm shown in Figure 6-6 runs steps 4 and 5, and iterates until one of the following satisfied: no classification error found, no noticeable performance improvement is achieved compared to previous iteration, or the maximum number of iterations is reached. In our implementation we limited the number of iterations to a maximum of 30, the performance ratio threshold used was 95% and, the learning ratio $\alpha = 0.1$. The details of the training experiments and testing are explained with the experimental work in the next section (Section 6.7).

6.7 Experimental work

In the context of this chapter we divided our experiments into two parts:

1. The performance evaluation of the Arabic phoneme training which included:

- Performance evaluation of Arabic phoneme dependent training.
- Performance evaluation of Arabic phoneme independent training.

2. Testing of the Arabic spoken utterance transcription based on the trained LVQ codebooks without phoneme bigram model.

6.7.1 Performance Evaluation of the Arabic Phoneme Training

The LVQ algorithm was applied on the Arabic phonemes. For each phoneme, we recorded; the number of frames belong to current phoneme and the number of complement frames (frames that belong to other phonemes rather than current one) , the sizes of the codebooks, the migrated frames from both sides (A or B) and the performance of the recognition.

6.7.1.1 Performance Evaluation of Arabic Phoneme Dependent Training

In this part of evaluation, 50k of sampled frames were used in training the codebooks. Both In-Class frames and Out-Class frames were considered in the training process. The performance of each phoneme based on dependent training is shown in Table 6-3. From Table 6-3, we can see that the weighted average performance for the phonemes after training was 90%.

Table 6-3: Evaluation of Phoneme-dependent classification

Phonemes	Number Of A Classified To B Error	Number Of B Classified To A Error	**Performance**
SIL	0	37	100.00%
+INH+	7	10	96.35%
AA:	92	116	79.51%
AH:	21	16	90.58%
IH	627	620	82.58%
IY	76	49	94.50%
IX	16	25	95.80%
AE	1535	1444	70.49%
AA	96	107	86.36%
AH	6	1	96.57%
UH	103	82	91.37%
UW	62	110	92.01%
UX	4	21	96.72%
AW	11	76	96.55%
E	566	522	74.24%
AY	13	4	94.82%
B	53	74	95.19%
T	110	77	95.00%
TH	58	58	87.45%
JH	29	22	95.52%
HH	23	12	95.89%
KH	23	20	95.43%
D	71	87	93.87%
DH	3	0	96.91%
R	127	89	91.19%
Z	16	10	96.84%
S	64	164	95.31%
SS	51	22	90.84%
SH	32	13	95.22%
DD	11	5	95.44%
TT	52	47	86.49%
DH2	3	0	96.70%
AI	50	87	96.25%
GH	2	1	97.80%
F	80	65	92.77%
Q	46	47	95.28%
K	36	55	95.11%
L	204	280	92.74%
M	124	91	94.10%
N	236	237	90.09%
H	182	205	85.89%
W	66	110	93.57%
Y	88	54	94.38%
AE:	312	495	85.44%
Overall Weighted Average			**90%**

6.7.1.2 Performance evaluation of Arabic phonemes

In the second part of evaluation, we used the same training set and combined the In-Class codebooks representatives only. Out-Class frames were ignored. The combined codebooks were then re-estimated using the LVQ learning algorithm. The phoneme independent performance results are shown in Table 6-4. The weighted average performance achieved from independent classification training was 98.49%, which is higher than the first experiment. The newly trained codebooks gained from independent classification will be used in the testing phase to do utterance transcription. The previous evaluations are summarized in Table 6-5.

Table 6-4: Phoneme-independent classification of the training set

Phonemes	Number Of A Classified To B Error	Number Of B Classified To A Error	**Performance**
SIL	0	37	100.00%
+INH+	7	10	97.40%
AA:	5	3580	98.89%
AH:	1	1668	99.55%
IH	41	12148	98.86%
IY	27	3906	98.04%
IX	5	1749	98.69%
AE	56	14808	98.92%
AA	4	3823	99.43%
AH	0	502	100.00%
UH	34	4312	97.15%
UW	16	3511	97.94%
UX	3	48	97.54%
AW	0	1328	100.00%
E	33	13447	98.50%
AY	6	850	97.61%
B	15	2497	98.64%
T	34	3111	98.45%
TH	13	3031	97.19%
JH	26	1457	95.99%
HH	8	132	98.57%
KH	7	195	98.61%
D	21	3290	98.19%
DH	1	38	98.97%
R	60	7905	95.84%
Z	8	180	98.42%
S	11	1493	99.19%
SS	18	906	96.77%
SH	1	440	99.85%
DD	11	1196	95.44%
TT	15	2656	96.10%
DH2	0	23	100.00%
AI	22	3421	98.35%
GH	0	13	100.00%
F	21	2840	98.10%
Q	7	2106	99.28%
K	10	1415	98.64%
L	29	10311	98.97%
M	10	2530	99.52%
N	20	4943	99.16%
H	56	8130	95.66%
W	66	110	93.57%
Y	16	2899	98.98%
AE:	7	6259	99.67%
Overall Weighted Average			**98.49%**

Table 6-5: Dependent/Independent Performance Evaluation

Num	Sample Size	Type	Achieved Performance
1	50 K	Training (Dependent Classification)	90 %
2	50 K	Training (Independent Classification)	98.49 %

6.8 Arabic Spoken Utterance Transcription Using LVQ Codebooks

To evaluate the usefulness of the LVQ algorithm for Arabic phoneme transcription, we applied the LVQ algorithm on the spoken utterance transcription. In this testing part, we depended on the trained codebooks resulted from independent classification. Then, we iterated on the test set utterances files. For each file we applied steps 1 to 4 of the algorithm shown in Figure 6-6. Figure 6-7, summarized the utterance transcription process, which can be described as follows:

The utterance transcription consists of the following steps:

1- Read the Arabic spoken utterance wave file with its predetermined transcription.

2- Convert the wave file to an MFCC feature vector file.

3- Preprocess the feature vector file by preparing its proper dimensionality.

4- Load the LVQ trained codebook for each of the Arabic phonemes and apply the following classification steps:

   a. For each frame in the utterance feature matrix, calculate the minimum Euclidian distance between the frame and the centers of all trained codebooks.

   b. Calculate the summation of the minimums in (a).

   c. For each trained codebook, evaluate the activation value given by the following formula [4]:

$$\mathbf{AV = 1 - \frac{Distance(Phoneme)}{Sum(Minimum\ Distances)}} \qquad (6\text{-}2)$$

d. Assign the current frame to the class of the corresponding phoneme of the maximum AV value.

The average recognition rate we have achieved at the phoneme level was 72% comprising the test set of the corpus. This ratio is justifiable because, neither HMM nor any phoneme bigram tables were used. The results gained from LVQ are comparable to that gained when using HMM but, without bigrams model. Details are shown in chapter 5, it was 56.7% of correctness.

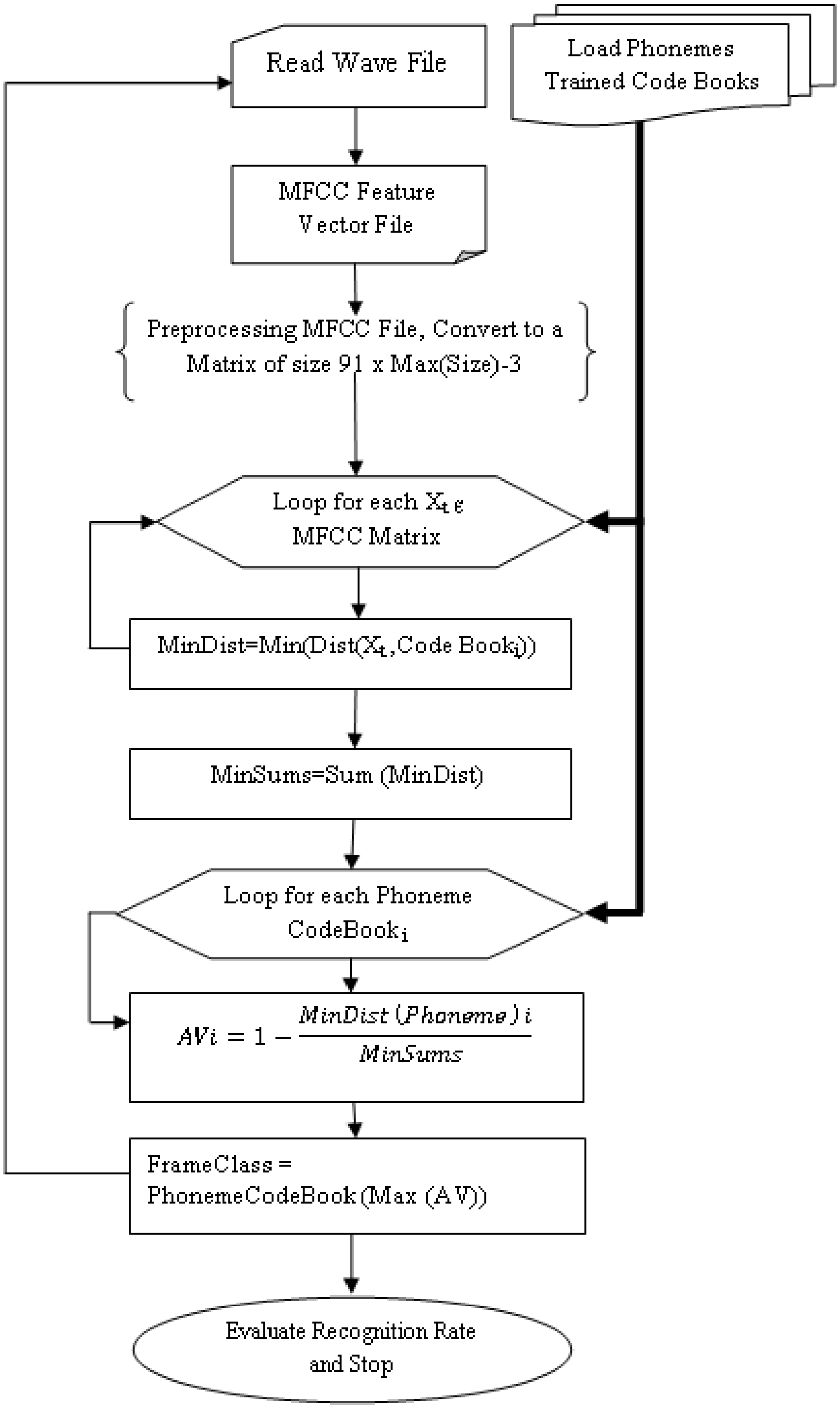

Read Wave File
Load Phonemes Trained Code Books
MFCC Feature Vector File
Preprocessing MFCC File, Convert to a Matrix of size 91 x Max(Size)-3
Loop for each $X_t \in$ MFCC Matrix
MinDist=Min(Dist(X_t,Code Book$_i$))
MinSums=Sum (MinDist)
Loop for each Phoneme CodeBook$_i$
$AVi = 1 - \frac{MinDist\ (Phoneme)\ i}{MinSums}$
FrameClass = PhonemeCodeBook (Max (AV))
Evaluate Recognition Rate and Stop

Figure 6-7: Utterance Phoneme Transcription flowchart

Table 6-6, provides a brief comparison of our work to the work mentioned in our literature review. Although the literature review contains many languages with many different corpuses using the LVQ, Our work is the one of few efforts invested in continuously open-vocabulary Arabic speech In many of other cases, the pronunciations are either for isolated words or for digits. Fewer research efforts concentrated on Arabic continuous speech. Also, the corpus we used consists of 5recorded hours of continuous Arabic speech, which is significantly bigger than the other corpora, to our best knowledge. Even we sampled 50K frames for training due to memory limitation; we have a K-means algorithm with splitting. We applied the testing on the entire test set in batches of 20K. Our results showed that we achieved the best performance ratio of 98.49% during codebook training of independent classification. In addition, we measured the recognition rate of utterance phonemes transcription and we achieved 72% without involving any phoneme bigram tables or any other learning models like HMM for example.

Table 6-6: Comparison of our work to the literature review

Work	Level/Language	Approach	Vocabulary	Corpus Used	Best Reported Accuracy
(McDermott and Katagiri, 1991)	Isolated Japanese Words	NN- architecture based on LVQ	Speaker-Dependent	5240 Common Japanese Words	98%-99%
(Avdagic et al., 2007)	Isolated Bosnian Words	Hybrid LVQ/Elman NN	Open with 30 Phonemes	Details not Available	Average of 98%
(Kondo et al., 1994)	Isolated and Continuously Spoken Japanese Digits	LVQ	Closed	Details not Available	99.2% for isolated digits 95.4% for continuously spoken digits
(Mäntysalo et al., 1994)	Isolated Finnish Words	LVQ and discrete HMM	Closed	311 Words Spoken by 3 Males, Repeated 4 Times.	99%
Avdagic, Z. et al. (Ghrgen et al., 1994)	Isolated Japanese Words	NN And LVQ	Speaker-Dependent	5240 Common Japanese Words	95.8%
(Cosi et al., 2000)	Continuous Italian Speech	NN And LVQ	Open	AIDA Standard Speech Database	67.9%
(Kurimo, 1997)	Continuous Finnish Speech	HMM with SOM and LVQ	Closed	Details not Available	94% Word Recognition Rate
(Essa et al., 2008b)	Isolated Arabic Words	NN And LVQ	Closed	Details not Available	96%
(Ma and Zeng, 2012)	Isolated Arabic Digits	Modified LVQ	Closed	Details not Available	94.21%
(Selouani and Caelen, 1999)	Continuous Arabic Speech	TDNN and LVQ	Open	Arabic Speech Corpus	87%
Our Work	Continuous Arabic Speech	LVQ	Open	Arabic Speech Corpus (4 Recorded Hours from TV-News)	98.49%

6.9 Chapter Summary

We have applied the LVQ algorithm on Arabic phoneme recognition. The algorithm showed a high rate of frame labeling performance, ranging between 90% and 98.49%. Our reported performance is in line with that of (McDermott and Katagiri, 1991) and as good as the results achieved by (Waibel et al., 1989). We also carried out an experiment for Arabic phoneme alignment (Recognition) using the trained codebooks of the Arabic phonemes. We achieved around 72% of accuracy without using any phoneme bigram model. The phoneme transcription usually depends on the phoneme bigrams model and most of the research done in this area depends on it. The purpose behind that is to achieve more accuracy. In our research, we achieved a reasonable accuracy without the use of the bigram models.

CHAPTER 7

ARABIC PHONEME RECOGNITION USING COMBINED LVQ AND HMM MODEL

7.1 INTRODUCTION

In this chapter, we present the use of the Learning Vector Quantization (LVQ) algorithm with Hidden Markov Model (HMM) as a hybrid model to recognize Arabic phonemes. An open-vocabulary continuous Arabic speech corpus of different TV news clips was used. We employed a data driven approach to generate the training feature vectors that embed the frame neighboring correlation information. Next, we generated the phonemes codebooks using the K-means splitting algorithm. After that, we trained the generated codebooks using the LVQ algorithm. We achieved a frame labeling performance of 98.49% during independent classification and 90% during dependent classification (See chapter 6). When using the trained LVQ codebooks in the utterance phoneme transcription, we reached phoneme recognition rate of 72% using LVQ only. In this chapter, we combined the LVQ codebooks with the single emitting state HMM model using enhanced Viterbi Algorithm that included the phonemes bigrams. We achieved 89% of recognition rate for the knowledge-based Arabic phonemes based on the newly LVQ/HMM algorithm.

7.2 Proposed Methodology

Our methodology in this chapter follows the same steps in chapter 6 in addition to some extra steps for combining the two models. The steps are as follows:

1. Data preparation phase.
2. Data clustering phase.
3. Training the codebooks using LVQ algorithm.
4. Performance evaluation of the phoneme recognition.
5. Combining the LVQ with HMM using Viterbi Algorithm
6. Performance Comparison of both standalone LVQ and Hybrid LVQ/HMM models.

Figure 7-1, describes the LVQ/HMM methodology and the steps added to Figure 6-1.

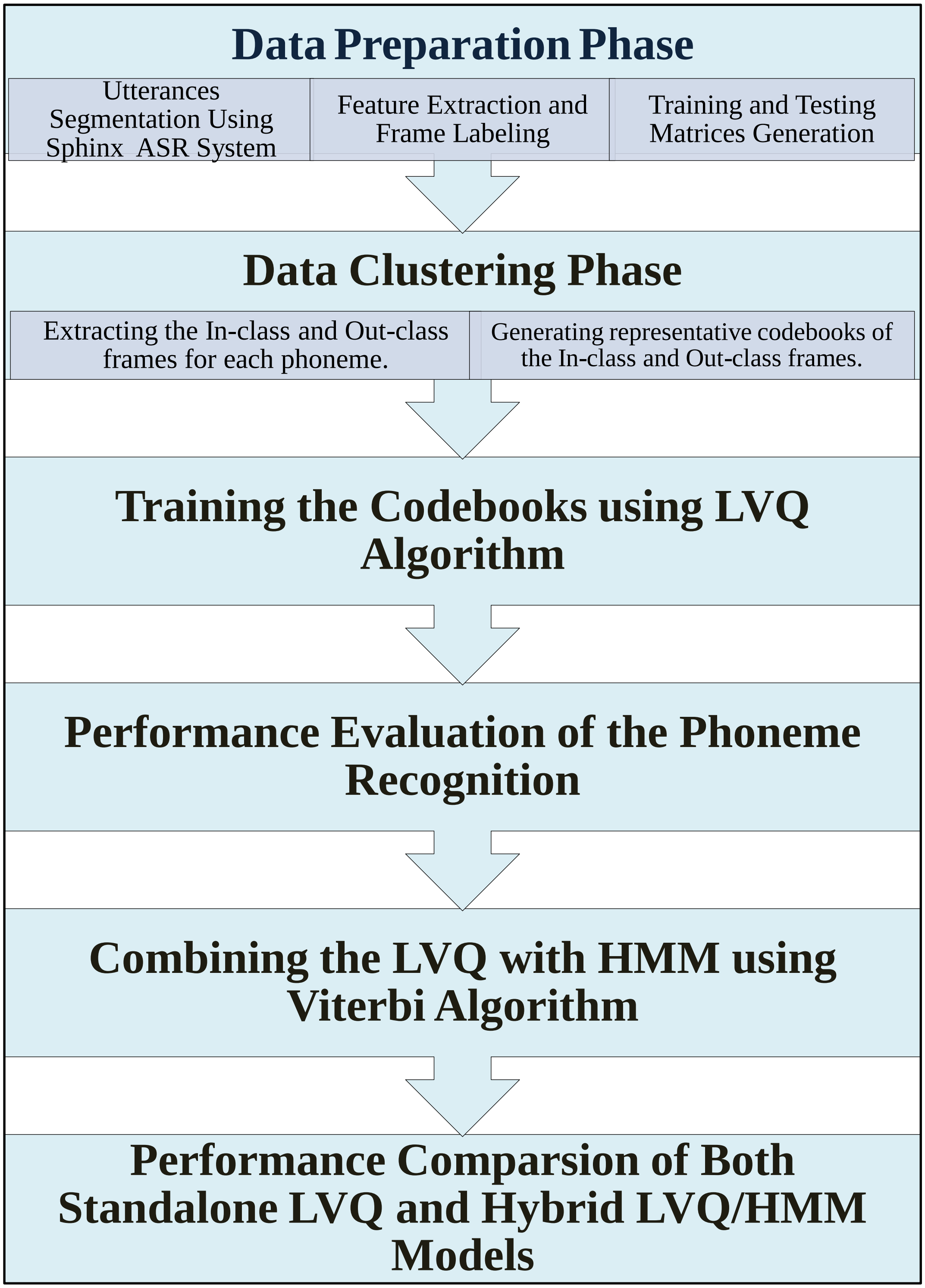

Data Preparation Phase
Utterances Segmentation Using Sphinx ASR System
Feature Extraction and Frame Labeling
Training and Testing Matrices Generation
Data Clustering Phase
Extracting the In-class and Out-class frames for each phoneme.
Generating representative codebooks of the In-class and Out-class frames.
Training the Codebooks using LVQ Algorithm
Performance Evaluation of the Phoneme Recognition
Combining the LVQ with HMM using Viterbi Algorithm
Performance Comparsion of Both Standalone LVQ and Hybrid LVQ/HMM Models

Figure 7-1: Arabic Phoneme recognition Methodology Phases

7.3 Combining LVQ and HMM

For each utterance in the testing set we apply the following procedure:

1. Extracting the feature vectors and corresponding phoneme sequence of the vectors.
2. Generating the 91 dimensional matrixes by the same process of generating the training matrix mentioned in section 6.3.1.2 with its corresponding label matrix.
3. Calculating the Activation Values (AV) for each phoneme based on the flow chart of Figure 6-7 and for each of the 91 dimensional frames. At this step the minimum distances of the 91 dimensional frames from each phoneme are calculated and stored.
4. Generated A matrix of size N by M for each utterance, where N is the number of phoneme codebooks (45 in our case) and M is the number of frames in the utterance.
5. Extracting the phoneme bigrams and the phoneme minimum and maximum durations from the corpus. The phoneme bigram in this case represents the state transition matrix of a phoneme.
6. Applying VA used in (Gemmeke et al., 2009) with some modifications (see next paragraphs) to find the most probable phoneme transcription of the utterance.

The N by M matrix represents the probability of a phoneme given the observation i.e. **P(Ph|O)** while the emitting probability matrix represents the probability of observation given a phoneme i.e. **P(O|Ph)** which can be calculated using the following equation (Rabiner, 1989):

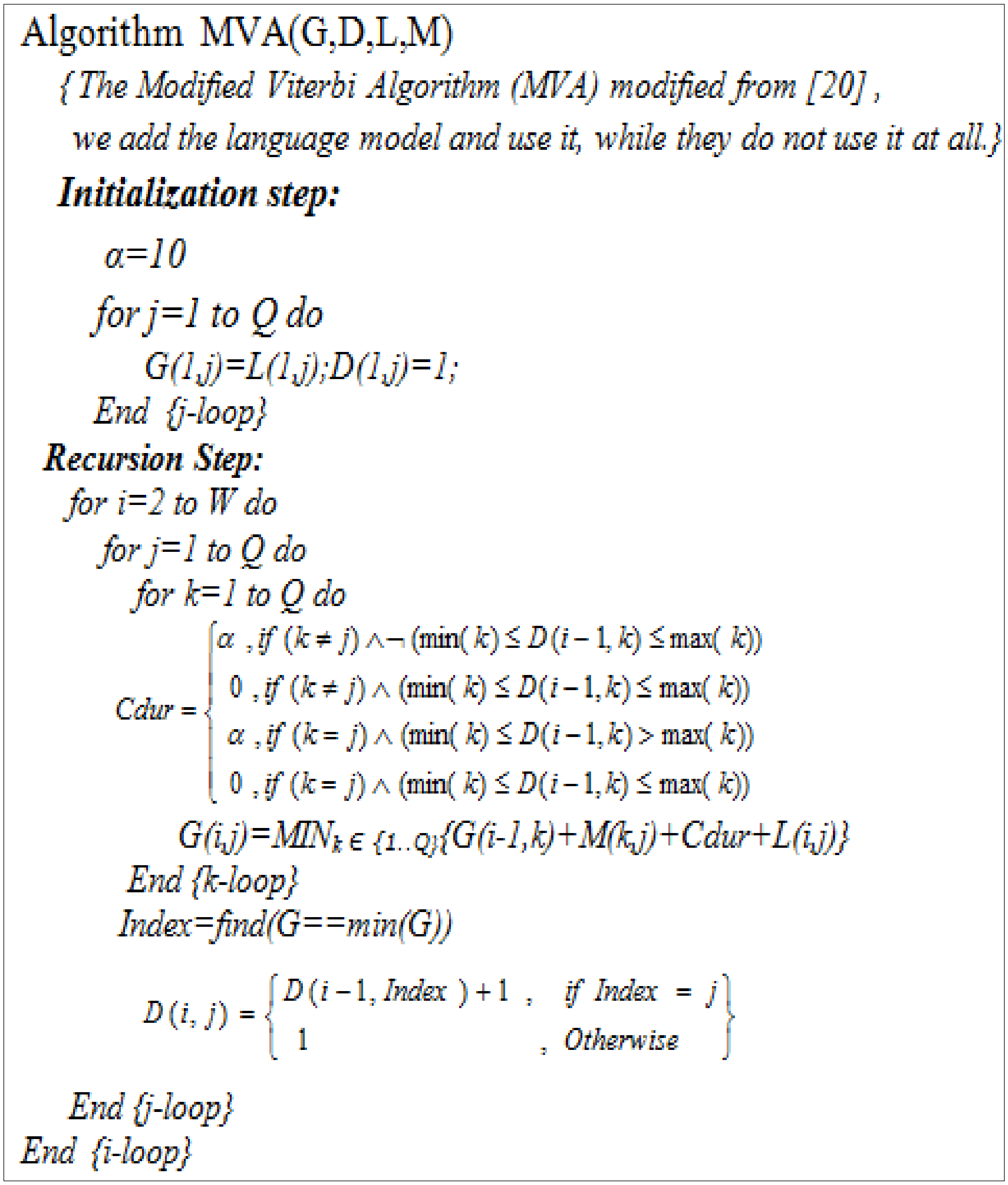
$$\mathbf{P(O|Ph)} = \frac{\mathbf{P(Ph|O)}}{\mathbf{P(Ph)}} \qquad (7\text{-}1)$$

Where **P(Ph|O)** is based on the minimum distances using LVQ and the **P(Ph)** (Probability of the phoneme) is extracted from the corpus itself, before applying Equation (7-1). The minimum Euclidean distances must be converted to probability by using the following equation (Rabiner, 1989):

$$\mathbf{Probability(i,j)} = \mathbf{e}^{\frac{-\mathbf{d(i,j)}^{2}}{2\sigma(\mathbf{j})^{2}}} \qquad (7\text{-}2)$$

Where **d(i,j)** is the Euclidean distance entry in the matrix and **σ** is the standard deviation of the column j.

Our modified VA used in our hybrid model is based on the one used in (Gemmeke et al., 2009) and described in Figure 7-2.



```
Algorithm MVA(G,D,L,M)
  {The Modified Viterbi Algorithm (MVA) modified from [20],
   we add the language model and use it, while they do not use it at all.}
  Initialization step:
      $\alpha=10$
      for j=1 to Q do
          G(1,j)=L(1,j);D(1,j)=1;
      End {j-loop}
 Recursion Step:
   for i=2 to W do
      for j=1 to Q do
        for k=1 to Q do
            $Cdur = \begin{cases} \alpha \text{ , if } (k \neq j) \wedge \neg(\min(k) \leq D(i-1,k) \leq \max(k)) \\ 0 \text{ , if } (k \neq j) \wedge (\min(k) \leq D(i-1,k) \leq \max(k)) \\ \alpha \text{ , if } (k = j) \wedge (\min(k) \leq D(i-1,k) > \max(k)) \\ 0 \text{ , if } (k = j) \wedge (\min(k) \leq D(i-1,k) \leq \max(k)) \end{cases}$
            $G(i,j)=MIN_{k \in \{1..Q\}}\{G(i-1,k)+M(k,j)+Cdur+L(i,j)\}$
        End {k-loop}
        Index=find(G==min(G))
        $D(i,j) = \begin{cases} D(i-1, Index)+1 \text{ , } & \text{if } Index = j \\ 1 & \text{, Otherwise} \end{cases}$
    End {j-loop}
End {i-loop}
```

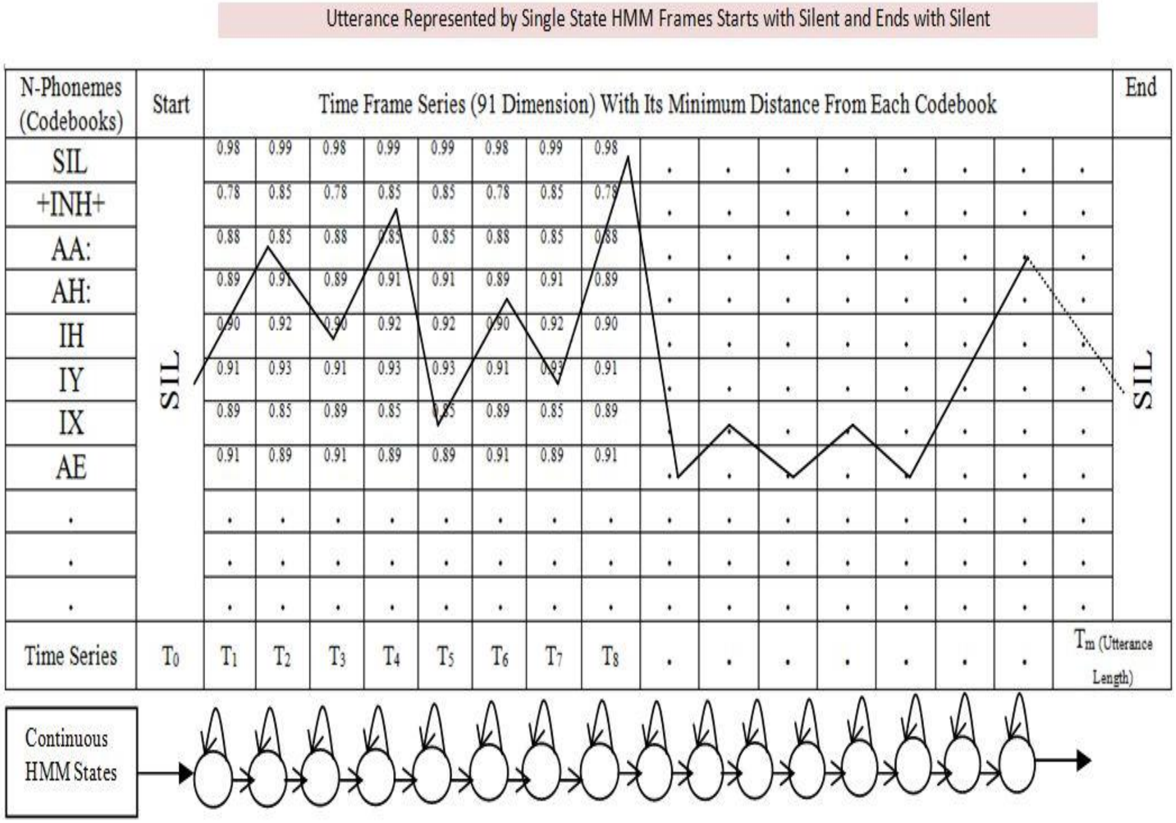
Figure 7-2: Our Modified Viterbi Algorithm Taken From (Gemmeke et al., 2009)

Since the state duration is critical to our model, the traditional VA will not work without adding the state duration. Many state duration paradigms were studied, but they were not adequate for our model, where state hanging phenomena is to be appeared. The parameters of the modified VA used in Figure 7-2 are explained in Table 7-1. The modified algorithm inserts the language model in the probability calculations while it was not included in (Gemmeke et al., 2009) version. As a result the structure of the algorithm had to be changed in order to get benefit from the state duration function. The state duration function limits the duration between minimum and maximum phoneme length in the corpus. We extract those limits from the corpus for each phoneme and put them in a database for the modified algorithm Figure 7-2.

Table 7-1: Modified Viterbi Algorithm Parameters

Parameter	Parameter Usage	Initialized By
L	Local Cost Matrix	L= - Emitting Matrix
G	Global Cost Function (Used Also in Conventional Viterbi)	G(1,j)=L(1,j)
D	D(i,j) is the duration of the most recent label hypothesis along the best path from first column to (i,j).	D(1,j)=1
M	Phoneme Bigram (Language Model)	Extracted From The Corpus
W	Utterance Length in Frames (m)	
Q	Number of Phonemes (n)	
α	Default Duration Constant	10
Cdur	Duration Function Depends on Minimum (min(k)) and Maximum (Max(k)) duration value of a Phoneme in The Corpus.	α
Index	Handel the minimum score index for backward purposes	

The next step is to apply the modified Viterbi Algorithm (MVA) on the model that is described in Figure 7-3. In Figure 7-3, the columns represent the minimum Euclidean distances where each column represents the score of a phoneme given the observation. These scores were converted to probability by equation (7-1) The angled line in Figure 7-3 represents the most probable path of the time frame generated by MVA. A silent phoneme is added at the beginning (we call it T_0) and at the end of each utterance. The frame in the model represents a single state HMM with duration limit for each phoneme. When entering a state the duration limit along with the transition probability (bigrams) determines the suitable time to exit to next state.


Utterance Represented by Single State HMM Frames Starts with Silent and Ends with Silent
Continuous HMM States

N-Phonemes (Codebooks)	Start	Time Frame Series (91 Dimension) With Its Minimum Distance From Each Codebook																End
SIL	SIL	0.98	0.99	0.98	0.99	0.99	0.98	0.99	0.98	.	.	.	.	.	.	.	.	SIL
+INH+		0.78	0.85	0.78	0.85	0.85	0.78	0.85	0.78	.	.	.	.	.	.	.	.	
AA:		0.88	0.85	0.88	0.85	0.85	0.88	0.85	0.88	.	.	.	.	.	.	.	.	
AH:		0.89	0.91	0.89	0.91	0.91	0.89	0.91	0.89	.	.	.	.	.	.	.	.	
IH		0.90	0.92	0.90	0.92	0.92	0.90	0.92	0.90	.	.	.	.	.	.	.	.	
IY		0.91	0.93	0.91	0.93	0.93	0.91	0.93	0.91	.	.	.	.	.	.	.	.	
IX		0.89	0.85	0.89	0.85	0.85	0.89	0.85	0.89	.	.	.	.	.	.	.	.	
AE		0.91	0.89	0.91	0.89	0.89	0.91	0.89	0.91	.	.	.	.	.	.	.	.	
.		.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.		.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
.		.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
Time Series	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	.	.	.	.	.	.	.	T_m (Utterance Length)	

Figure 7-3: Viterbi Most Probable Path (LVQ/HMM) Mode

After applying our model on one thousand utterances in the test set, the average phoneme accuracy achieved was 89%. Figure 7-4 compare between the phoneme accuracy for the standalone LVQ and the hybrid model of LVQ/HMM.

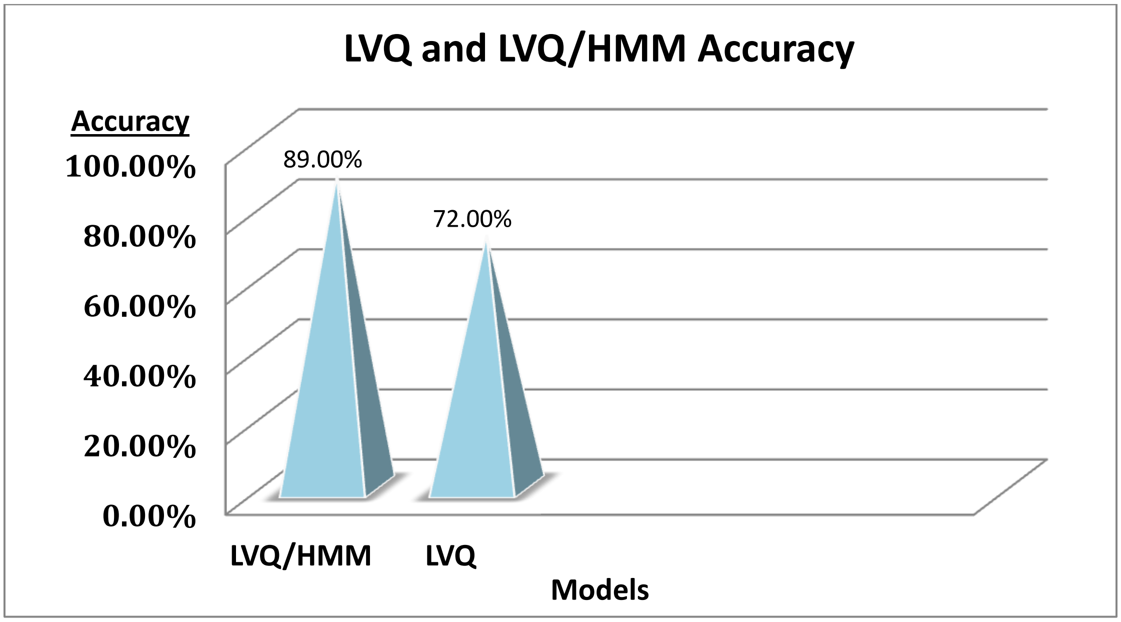


Figure 7-4: Comparison Of LVQ and LVQ/HMM Accuracy on Phoneme-Level Transcription

7.4 Chapter Summary and Conclusion

In this chapter we combined the LVQ with the HMM in a model where the phoneme bigrams are included. A continuous single emitting state HMM model was also embedded in our model. The average phoneme transcription accuracy achieved was 89%.

We modified the VA used in (Gemmeke et al., 2009) to introduce the state duration information in our model and call it MVA. The MVA is very sensitive to state duration which is represented by a state duration function. This sensitivity made it possible to increase the phoneme transcription accuracy by replacing the current state duration function with another improved and accurate one. Seeking for better duration function becomes an open issue in this research.

Table 7-2 summarizes the results of different methodologies and algorithms used in chapter 5 through chapter 7 and the achieved accuracies. It also shows whether we used the bigram table or not during the phoneme recognition.

Table 7-2 : Summary of Arabic Phoneme Transcription Methods.

Method	HMM-states	Without Bigrams	With Bigrams	Accuracy
Arabic Phonemes Transcription Using Varying Number of HMM	Fixed (3)	**Yes**		43.3%
	Varying	**Yes**		56.7%
	Varying		**Yes**	96.3%
Arabic Phoneme Transcription Using Only LVQ	No HMM	**Yes**		72%
Arabic Phoneme Transcription Using LVQ/HMM	1 State for all		**Yes**	89%

CHAPTER 8

DATA-DRIVEN ARABIC SUB-WORD EXTRACTION USING LVQ

8.1 INTRODUCTION

This chapter focused on the problem of the automatic SWUs extraction for continuous Arabic speech recognition along with some other related problems. The steps we have followed to extract SWUs are discussed in this chapter, which is divided into the following sections: section 8.2 which describe the corpus used, section 8.3 where we start segmenting speech, section 8.4 was responsible for feature vectors extraction, section 8.5 where we used the K-Means for clustering the feature vectors , section 8.6 for labeling frames, section 8.7 for generating LVQ codebooks, section 8.8 where the transcription of the SWUs using LVQ starts, section 8.9 where the bigrams and probabilities of the SWUs extracted, section 8.10 where the emitting probability matrix , section 8.11where the LVQ/HMM are combined using VA and 8.12 summarizes our chapter.

These steps may iterate until the required sub-word unit transcription accuracy is reached. The coming sections explain every step in details.

8.2 The Corpus

The corpus we have used is the one we have described in CHAPTER 4 section 4.2.

8.3 Speech Segmentation

The automatic speech segmentation is defined as breaking the continuous sound streams into discrete and non-overlapping basic units like words, syllables and phonemes (Rahman et al., 2012). These units could be recognized more easily than the whole stream. Speech segmentation is necessary for speech recognition (Kvale, 1993). The most two common methods for segmentation are: blind segmentation and aided segmentation (Sharma and Mammone, 1996). The blind segmentation algorithm is usually applied when no external knowledge is added to the segmentation algorithm. The blind segmentation algorithm is useful in speech recognition, speech segmentation and labeling, and in speaker verification system (Sharma and Mammone, 1996). Blind segmentation has two phases. The first phase depends on the acoustic features in the sound wave itself, while the second phase is built on a front-end parameterization of the speech signal by using Mel-Frequency Cepstral Coefficients (MFCC) or pure (Fast Fourier Transformation) FFT (SaiJayram et al., 2002).

The aided segmentation uses external linguistic knowledge like the orthographic and phoneme transcriptions in parallel with the input speech. The most common aided method for annotating phonemes in ASR systems is HMM-based annotating system (Juang and Rabiner, 2004). Figure 8-1 shows a suggested segmentation and labeling of the sentence "**بسم الله الرحمن الرحيم**" recorded through Audacity software.

In our research we have used a corpus of 5 recorded hours for MSA (See Section 4.2), the recorded hours were passed to sphinx ASR system for segmentation using HMM model with knowledgebase phoneme set. A phoneme level transcription and a word-level transcription were generated by sphinx forced-alignment mechanism. Table 4-2 shows a sample of the word and phoneme level segmentation using sphinx ASR system. In addition, the MFCC files correspond to the WAV files were also generated. In previous chapters, we used a predefined set of phonemes defined by expert linguistics, and then we applied our methodology. In this chapter we used only the WAV and MFCC files to derive the SWUs instead of the predefined set of phonemes. Moreover, the dictionary and utterances transcriptions were provided by the corpus. The next section will explain features extraction.

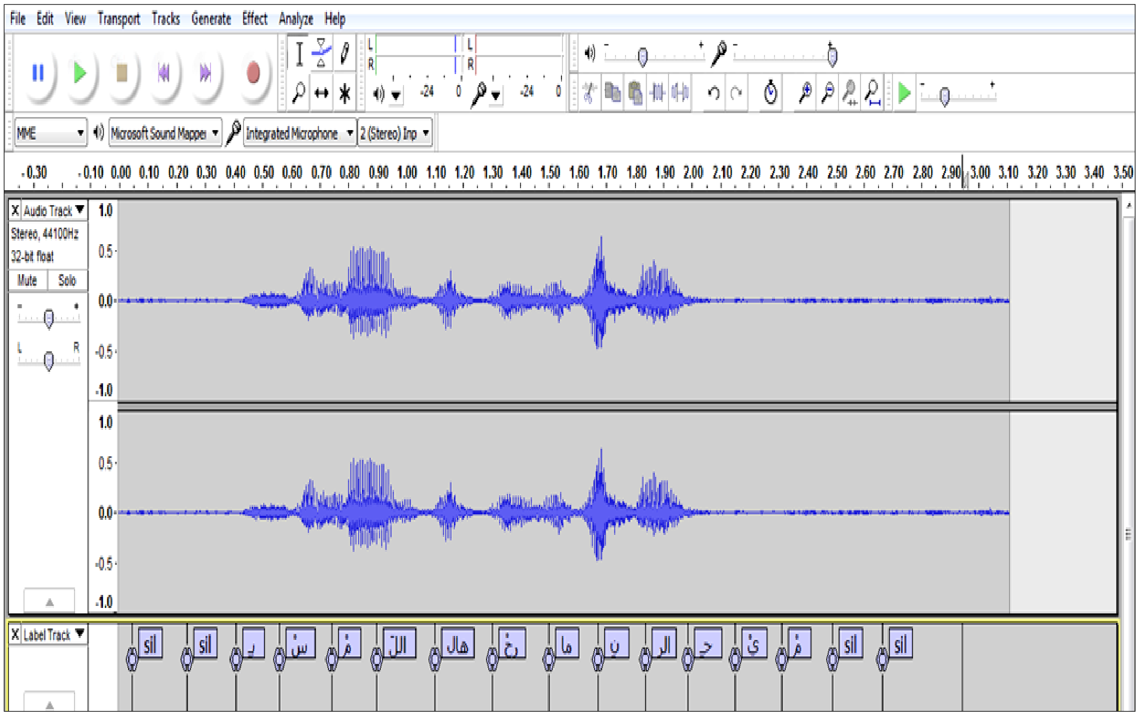
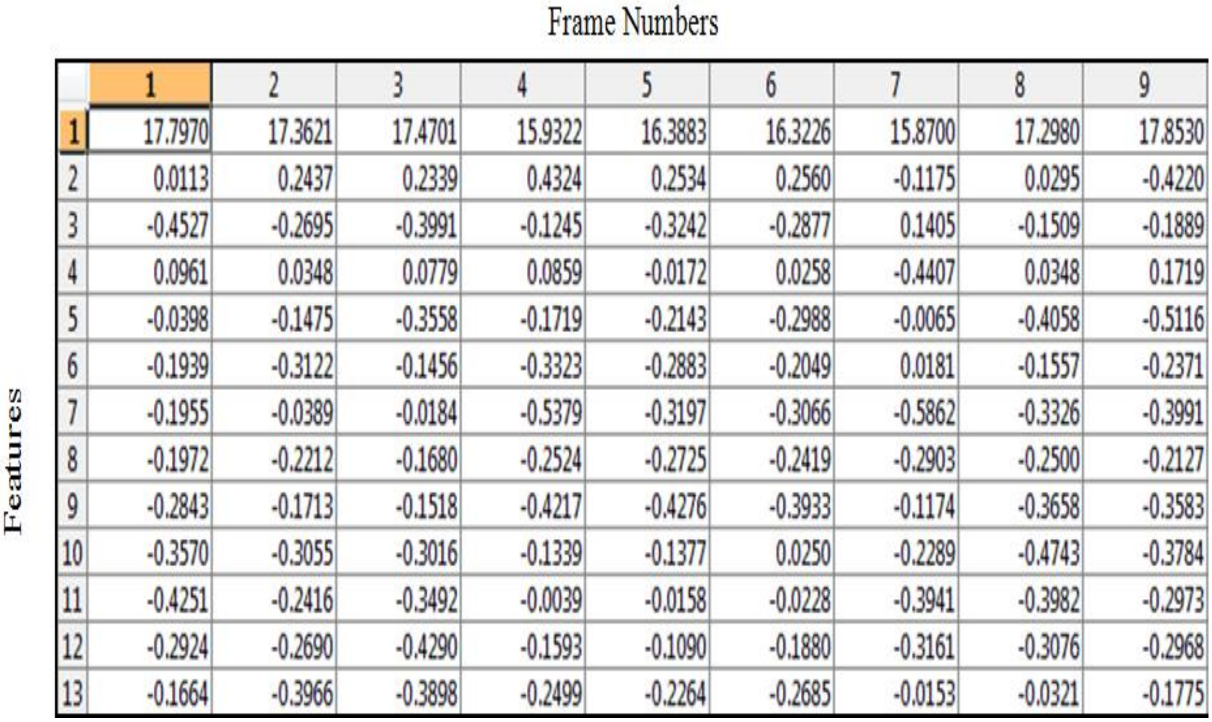


Figure 8-1: Segmentation of "بسم الله الرحمن الرحيم" Using Audacity

8.4 Feature Extraction

Since we already have the WAV files and corresponding MFCC files from the corpus, we developed a MATLAB code for reading MFCC files as feature vectors and store them in a matrix. Every recorded utterance in the training set has unique MFCC file. Dividing the first value in the MFCC file of an utterance by 13 results in the total number of frames for the utterance, where each frame consists of 13 features. The first value in the MFCC file stored as integer 32 in IEEE-be format (Big Indian Format) and the rest of values (Frames) stored as floating point in IEEE-be format. We passed over all the MFCC files in the training set and read them all in order and stored them in one matrix called *training feature matrix*. The feature vectors of already recorded utterances in Arabic language were generated. A population of feature vectors was generated for all the training set files. Figure 8-2 shows a snapshot of the feature vector matrix.



	1	2	3	4	5	6	7	8	9
1	17.7970	17.3621	17.4701	15.9322	16.3883	16.3226	15.8700	17.2980	17.8530
2	0.0113	0.2437	0.2339	0.4324	0.2534	0.2560	-0.1175	0.0295	-0.4220
3	-0.4527	-0.2695	-0.3991	-0.1245	-0.3242	-0.2877	0.1405	-0.1509	-0.1889
4	0.0961	0.0348	0.0779	0.0859	-0.0172	0.0258	-0.4407	0.0348	0.1719
5	-0.0398	-0.1475	-0.3558	-0.1719	-0.2143	-0.2988	-0.0065	-0.4058	-0.5116
6	-0.1939	-0.3122	-0.1456	-0.3323	-0.2883	-0.2049	0.0181	-0.1557	-0.2371
7	-0.1955	-0.0389	-0.0184	-0.5379	-0.3197	-0.3066	-0.5862	-0.3326	-0.3991
8	-0.1972	-0.2212	-0.1680	-0.2524	-0.2725	-0.2419	-0.2903	-0.2500	-0.2127
9	-0.2843	-0.1713	-0.1518	-0.4217	-0.4276	-0.3933	-0.1174	-0.3658	-0.3583
10	-0.3570	-0.3055	-0.3016	-0.1339	-0.1377	0.0250	-0.2289	-0.4743	-0.3784
11	-0.4251	-0.2416	-0.3492	-0.0039	-0.0158	-0.0228	-0.3941	-0.3982	-0.2973
12	-0.2924	-0.2690	-0.4290	-0.1593	-0.1090	-0.1880	-0.3161	-0.3076	-0.2968
13	-0.1664	-0.3966	-0.3898	-0.2499	-0.2264	-0.2685	-0.0153	-0.0321	-0.1775

Figure 8-2: First 9 Vectors of the Feature Vector Matrix for the Training Set

8.5 Clustering

A MATLAB code is written to implement the K-means clustering algorithm. We used the K-means code for clustering the frames of the *feature vector matrix* extracted in section 8.4. We suggested a number of centers for the K-means clustering algorithm. These centers represent the SWUs that will be used in speech recognition after training them using LVQ. Initially, we started by 70 centers as a suggested number of SWUs. We run the K-means MATLAB code to extract these centers. The default Euclidean distance was used. It is worth mentioning that we investigated the use of other types of distance measurements like: Correlation, Hamming, Cityblock and Cosine distances for best representative centers but, we f found that the Euclidean distance is the only who gave the minimum error regarding our problem. After extracting the centers we assigned them arbitrary labels. The resulting centers and their corresponding labels were stored for further use.

8.6 Frames Labeling

The *feature vector matrix* was relabeled based on the newly generated centers. Knowing the newly transcription of each utterance is done based on the boundary of the utterance that we already know from sphinx utterance segmentation. Since we know the order of utterances taken and length of each one and its name, we can match between features in feature vector matrix and extract the newly labels of each utterance. Figure 8-3 shows how a frame corresponds to labels and corresponds to utterances. The sequences and lengths of utterances are known so we can know the new labels of each utterance. Moreover we can know the label of each word in an utterance since the word segmentation of each utterance is also known to us from sphinx word segmentation. Knowing the word labeling will be beneficial in dictionary generation. The feature vectors of 3428 files representing the training utterances were extracted and fed to the K-means clustering algorithm to cluster and label them for the further processing by the LVQ.

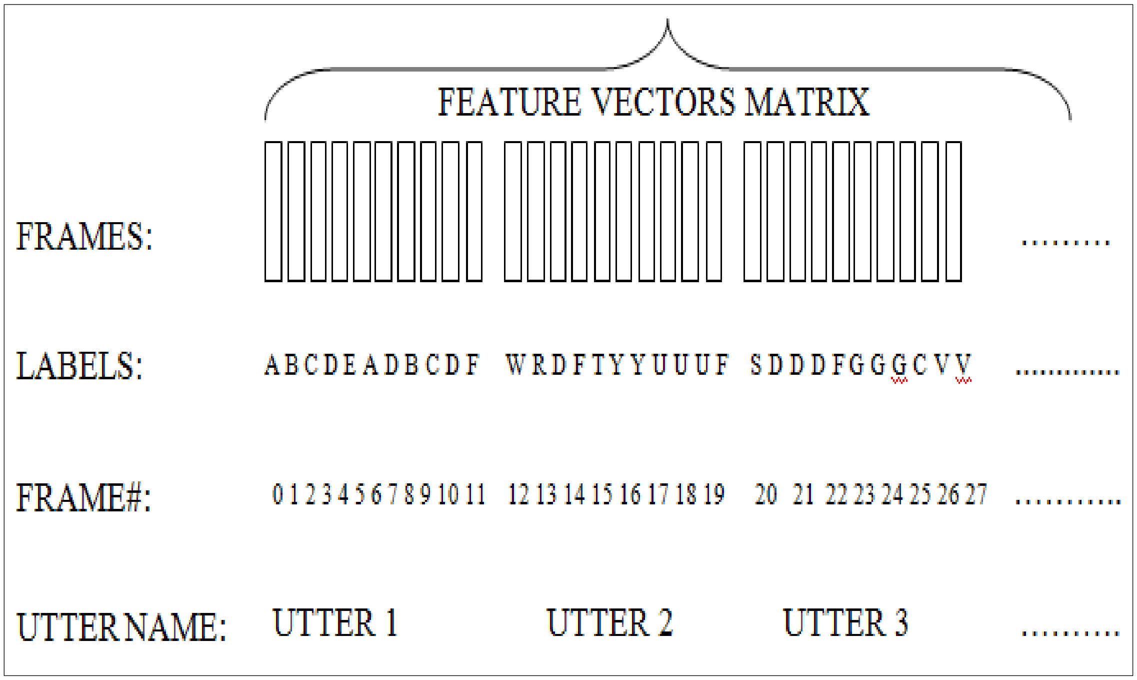


Figure 8-3: Extracting Utterances Labeling from K-means Clustering

8.7 Codebook Generation Using LVQ and Performance Evaluation

We followed the same steps mentioned in chapter 6 for generating the codebooks of the K-means selected centers, the whole data (3428 utterances) is taken into account and we did not reformulate the feature matrix to embed the neighboring correlation. We used the LVQ in this context for retraining multiple times until minimum errors reached, to force the right label for each frame so that we improve K-means labeling. Moreover, we have increased the number of iterations in K-means to reduce the mean square error. A MATLAB code was written to extract and train the codebooks based on LVQ algorithm.

The MATLAB code follows the steps of the algorithm; we divided it into 4 parts. The first part was responsible for extracting all frames and assigns them either 1 if they belong to the center or zero otherwise, for further use. The second part was responsible for determining the proper codebook size. Then, it uses the function of binary splitting to return the codebook as the centroids of the cluster. The third part of the LVQ program was the adaptation part of the codebooks, where this part of the code iterates for 50 iterations and at each iteration, it passed over the frames (the whole frames) one by one and adapt each frame position either to be in A or B classes. Finally, the fourth part of the MATLAB code was to calculate errors and training performance. We focus on the frames that migrate from A-class to B-class, because they are originally considered in A-class. This was our measure of accuracy. Accuracy is measured every iteration by the formula defined in equation 6-1, we exit the adaptation iteration loop either when the accuracy reaches 99% or above, or when we reach the maximum number of iterations, which is 50 in our case. During iterations, the learning ratio (α) is reduced by 85% to improve learning through the iterative process. The whole procedure is repeated for each sub-word until we have a set of trained codebooks represent the whole SWUs. For each SWU, we record and save; errors, performance, codebook sizes, and accuracy achieved. We started by 70 centers so we had 70 trained codebooks.

Table 8-1 shows the training performance results using LVQ, for 96% of accuracy while adapting the codebooks. Our target is to reach 99% or the minimum possible error for labeling alignment. Getting the best accuracy for the codebooks; means that we have the best labeling or the improved version of K-means labeling. The next step was to build the dictionary using those improved codebooks which represent the SWUs. The dictionary will be used in the recognition when combining the results from the LVQ with HMM definition using the LVQ. The next section discusses this issue in details.

Table 8-1: Training Performance for 150 SWUs Codebooks

Sub-Word Units	Number Of A-Centers	Number Of B-Centers	Number Of A Classified To B Error	Number Of B Classified To A Error	**Accuracy**
Aa	16	128	13	101	0.972860125
Ab	16	128	10	35	0.9609375
Ac	16	128	18	6	0.95862069
Ad	32	128	30	76	0.95215311
Ae	16	128	7	6	0.975945017
Af	16	128	5	9	0.987623762
Ag	16	128	8	19	0.968992248
Ah	32	128	15	16	0.971098266
Ai	16	128	19	85	0.956321839
Aj	16	128	7	5	0.974074074
Ak	16	128	9	16	0.960526316
Al	16	128	12	11	0.968337731
Am	16	128	17	5	0.955613577
An	32	128	23	149	0.961730449
Ao	16	128	18	23	0.952631579
Ap	32	128	32	67	0.951070336
Aq	16	128	11	29	0.967930029
Ar	16	128	11	4	0.955284553
As	32	128	27	2	0.955
At	16	128	14	11	0.952380952
Au	16	128	8	7	0.953216374
Av	16	128	24	37	0.951612903
Aw	16	128	15	45	0.956395349
Ax	32	128	45	46	0.951033732
Ay	32	128	26	23	0.95709571
Az	16	128	22	39	0.952173913
Ba	32	128	21	31	0.964824121
Bb	16	128	11	41	0.958646617
Bc	16	128	10	17	0.95215311

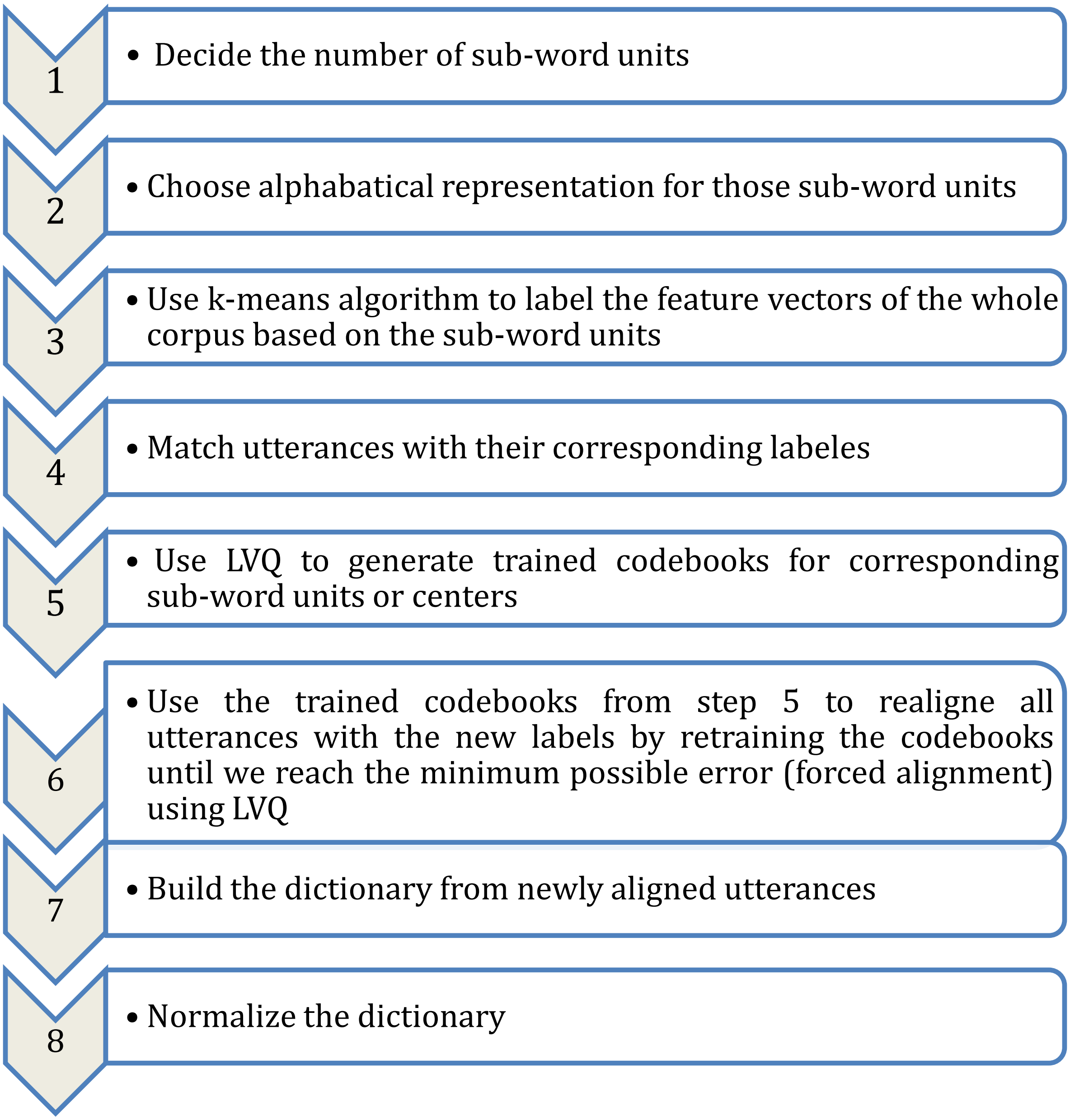
Sub-Word Units	Number Of A-Centers	Number Of B-Centers	Number Of A Classified To B Error	Number Of B Classified To A Error	**Accuracy**
Bd	16	128	7	7	0.978593272
Be	16	128	14	41	0.957831325
Bf	16	128	9	18	0.964566929
Bg	16	128	12	5	0.966666667
Bh	16	128	14	9	0.954983923
Bi	16	128	1	21	0.991935484
Bj	16	128	5	15	0.971264368
Bk	16	128	7	20	0.969162996
Bl	32	128	24	7	0.954285714
Bm	16	128	20	9	0.9543379
Bn	16	128	8	40	0.976331361
Bo	16	128	4	1	0.968
Bp	16	128	7	12	0.969162996
Bq	16	128	18	10	0.961620469
Br	32	128	28	17	0.950704225
Bs	16	128	21	21	0.954446855
Bt	16	128	15	29	0.962121212
Bu	16	128	8	9	0.958762887
Bv	16	128	23	27	0.953252033
Bw	16	128	12	8	0.961290323
Bx	16	128	21	79	0.95766129
By	16	128	8	12	0.963963964
Bz	16	128	15	23	0.951768489
Ca	16	128	11	6	0.963210702
Cb	16	128	1	20	0.989010989
Cc	16	128	11	40	0.958333333
Cd	16	128	22	26	0.953974895
Ce	16	128	14	10	0.96803653
Cf	16	128	19	18	0.960251046
Cg	16	128	19	67	0.956018519

Sub-Word Units	Number Of A-Centers	Number Of B-Centers	Number Of A Classified To B Error	Number Of B Classified To A Error	**Accuracy**
Ch	16	128	13	13	0.955631399
Ci	16	128	6	55	0.976923077
Cj	16	128	9	18	0.965116279
Ck	16	128	20	33	0.957894737
Cl	16	128	9	27	0.965384615
Cm	32	128	27	25	0.953846154
Cn	16	128	16	19	0.955182073
Co	16	128	18	15	0.950549451
Cp	16	128	13	14	0.961538462
Cq	16	128	5	11	0.96
Cr	16	128	19	32	0.953658537
Weighted Average					0.961200695

8.8 Dictionary Generation

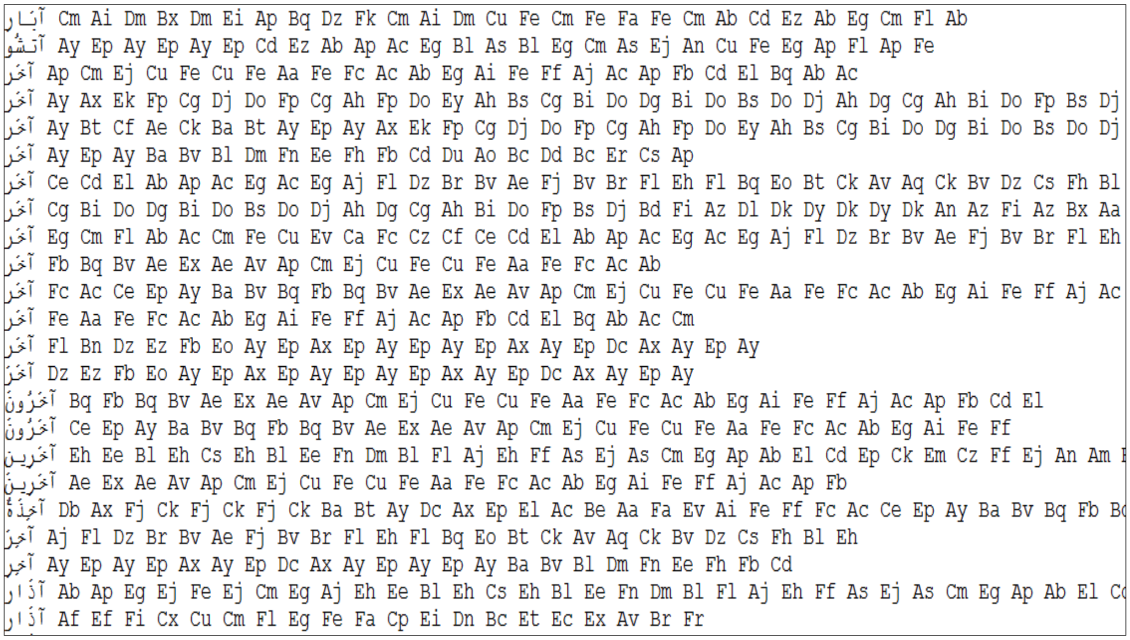
A new dictionary was built from the new frame-level labeling using SWUs or suggested centers. All the steps included in building the dictionary are shown in Figure 8-4.

As we see from Figure 8-4, the dictionary depends on both K-Means and LVQ algorithms. The centers or suggested SWUs are determined using K-means and are trained using the LVQ. Utterances were labeled based on trained codebooks that were extracted using LVQ, and the dictionary generated from the utterances once again. After that, the dictionary was normalized by two steps: numbering different pronunciations of words, and sorting data. The utterance SWUs alignment performed using LVQ alignment flow chart in Figure 6-7. No language model or any external knowledge was engaged in the process of labeling. Two types of dictionaries were produced, one with repeated consecutive SWUs, and one without repetition. Figure 8-5 and Figure 8-6 show parts of the generated dictionary with and without SWUs repetitions.



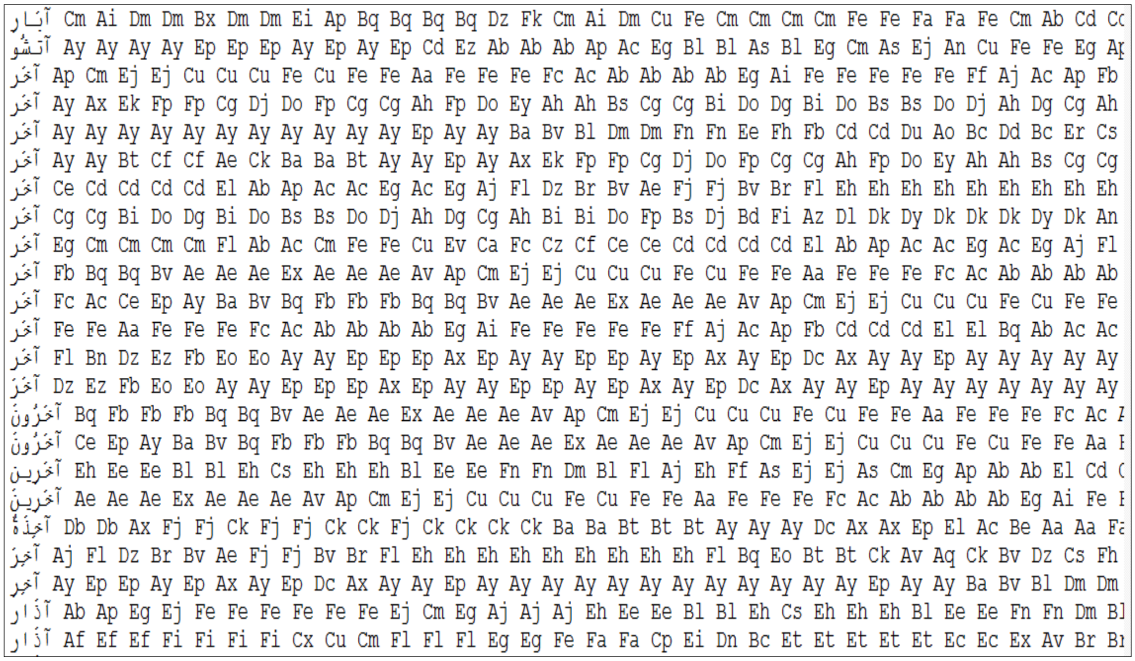
1. Decide the number of sub-word units
2. Choose alphabatical representation for those sub-word units
3. Use k-means algorithm to label the feature vectors of the whole corpus based on the sub-word units
4. Match utterances with their corresponding labeles
5. Use LVQ to generate trained codebooks for corresponding sub-word units or centers
6. Use the trained codebooks from step 5 to realigne all utterances with the new labels by retraining the codebooks until we reach the minimum possible error (forced alignment) using LVQ
7. Build the dictionary from newly aligned utterances
8. Normalize the dictionary

Figure 8-4: Dictionary Generation Process



```
آبَار Cm Ai Dm Bx Dm Ei Ap Bq Dz Fk Cm Ai Dm Cu Fe Cm Fe Fa Fe Cm Ab Cd Ez Ab Eg Cm Fl Ab
آتَشُو Ay Ep Ay Ep Ay Ep Cd Ez Ab Ap Ac Eg Bl As Bl Eg Cm As Ej An Cu Fe Eg Ap Fl Ap Fe
آخَر Ap Cm Ej Cu Fe Cu Fe Aa Fe Fc Ac Ab Eg Ai Fe Ff Aj Ac Ap Fb Cd El Bq Ab Ac
آخَر Ay Ax Ek Fp Cg Dj Do Fp Cg Ah Fp Do Ey Ah Bs Cg Bi Do Dg Bi Do Bs Do Dj Ah Dg Cg Ah Bi Do Fp Bs Dj
آخَر Ay Bt Cf Ae Ck Ba Bt Ay Ep Ay Ax Ek Fp Cg Dj Do Fp Cg Ah Fp Do Ey Ah Bs Cg Bi Do Dg Bi Do Bs Do Dj
آخَر Ay Ep Ay Ba Bv Bl Dm Fn Ee Fh Fb Cd Du Ao Bc Dd Bc Er Cs Ap
آخَر Ce Cd El Ab Ap Ac Eg Ac Eg Aj Fl Dz Br Bv Ae Fj Bv Br Fl Eh Fl Bq Eo Bt Ck Av Aq Ck Bv Dz Cs Fh Bl
آخَر Cg Bi Do Dg Bi Do Bs Do Dj Ah Dg Cg Ah Bi Do Fp Bs Dj Bd Fi Az Dl Dk Dy Dk Dy Dk An Az Fi Az Bx Aa
آخَر Eg Cm Fl Ab Ac Cm Fe Cu Ev Ca Fc Cz Cf Ce Cd El Ab Ap Ac Eg Ac Eg Aj Fl Dz Br Bv Ae Fj Bv Br Fl Eh
آخَر Fb Bq Bv Ae Ex Ae Av Ap Cm Ej Cu Fe Cu Fe Aa Fe Fc Ac Ab
آخَر Fc Ac Ce Ep Ay Ba Bv Bq Fb Bq Bv Ae Ex Ae Av Ap Cm Ej Cu Fe Cu Fe Aa Fe Fc Ac Ab Eg Ai Fe Ff Aj Ac
آخَر Fe Aa Fe Fc Ac Ab Eg Ai Fe Ff Aj Ac Ap Fb Cd El Bq Ab Ac Cm
آخَر Fl Bn Dz Ez Fb Eo Ay Ep Ax Ep Ay Ep Ay Ep Ax Ay Ep Dc Ax Ay Ep Ay
آخَرَ Dz Ez Fb Eo Ay Ep Ax Ep Ay Ep Ay Ep Ax Ay Ep Dc Ax Ay Ep Ay
آخَرُونَ Bq Fb Bq Bv Ae Ex Ae Av Ap Cm Ej Cu Fe Cu Fe Aa Fe Fc Ac Ab Eg Ai Fe Ff Aj Ac Ap Fb Cd El
آخَرُونَ Ce Ep Ay Ba Bv Bq Fb Bq Bv Ae Ex Ae Av Ap Cm Ej Cu Fe Cu Fe Aa Fe Fc Ac Ab Eg Ai Fe Ff
آخَرِين Eh Ee Bl Eh Cs Eh Bl Ee Fn Dm Bl Fl Aj Eh Ff As Ej As Cm Eg Ap Ab El Cd Ep Ck Em Cz Ff Ej An Am
آخَرِينَ Ae Ex Ae Av Ap Cm Ej Cu Fe Cu Fe Aa Fe Fc Ac Ab Eg Ai Fe Ff Aj Ac Ap Fb
آخِذَةٌ Db Ax Fj Ck Fj Ck Fj Ck Ba Bt Ay Dc Ax Ep El Ac Be Aa Fa Ev Ai Fe Ff Fc Ac Ce Ep Ay Ba Bv Bq Fb
آخِرَ Aj Fl Dz Br Bv Ae Fj Bv Br Fl Eh Fl Bq Eo Bt Ck Av Aq Ck Bv Dz Cs Fh Bl Eh
آخِر Ay Ep Ay Ep Ax Ay Ep Dc Ax Ay Ep Ay Ep Ay Ba Bv Bl Dm Fn Ee Fh Fb Cd
آذَار Ab Ap Eg Ej Fe Ej Cm Eg Aj Eh Ee Bl Eh Cs Eh Bl Ee Fn Dm Bl Fl Aj Eh Ff As Ej As Cm Eg Ap Ab El
آذَار Af Ef Fi Cx Cu Cm Fl Eg Fe Fa Cp Ei Dn Bc Et Ec Ex Av Br Fr
```

Figure 8-5: **Part of The Dictionary Labeled by Sub-Word Units without Repetitions**



```
آبَـار Cm Ai Dm Dm Bx Dm Dm Ei Ap Bq Bq Bq Bq Dz Fk Cm Ai Dm Cu Fe Cm Cm Cm Cm Fe Fe Fa Fa Fe Cm Ab Cd C
آتـشُو Ay Ay Ay Ay Ep Ep Ep Ay Ep Ay Ep Cd Ez Ab Ab Ab Ap Ac Eg Bl Bl As Bl Eg Cm As Ej An Cu Fe Fe Eg A
آخَر Ap Cm Ej Ej Cu Cu Cu Fe Cu Fe Fe Aa Fe Fe Fe Fc Ac Ab Ab Ab Ab Eg Ai Fe Fe Fe Fe Fe Ff Aj Ac Ap Fb
آخَر Ay Ax Ek Fp Fp Cg Dj Do Fp Cg Cg Ah Fp Do Ey Ah Ah Bs Cg Cg Bi Do Dg Bi Do Bs Bs Do Dj Ah Dg Cg Ah
آخَر Ay Ay Ay Ay Ay Ay Ay Ay Ay Ay Ay Ep Ay Ay Ba Bv Bl Dm Dm Fn Fn Ee Fh Fb Cd Cd Du Ao Bc Dd Bc Er Cs
آخَر Ay Ay Bt Cf Cf Ae Ck Ba Ba Bt Ay Ay Ep Ay Ax Ek Fp Fp Cg Dj Do Fp Cg Cg Ah Fp Do Ey Ah Ah Bs Cg Cg
آخَر Ce Cd Cd Cd Cd El Ab Ap Ac Ac Eg Ac Eg Aj Fl Dz Br Bv Ae Fj Fj Bv Br Fl Eh Eh Eh Eh Eh Eh Eh Eh Eh
آخَر Cg Cg Bi Do Dg Bi Do Bs Bs Do Dj Ah Dg Cg Ah Bi Bi Do Fp Bs Dj Bd Fi Az Dl Dk Dy Dk Dk Dk Dy Dk An
آخَر Eg Cm Cm Cm Cm Fl Ab Ac Cm Fe Fe Cu Ev Ca Fc Cz Cf Ce Ce Cd Cd Cd Cd El Ab Ap Ac Ac Eg Ac Eg Aj Fl
آخَر Fb Bq Bq Bv Ae Ae Ae Ex Ae Ae Ae Av Ap Cm Ej Ej Cu Cu Cu Fe Cu Fe Fe Aa Fe Fe Fe Fc Ac Ab Ab Ab Ab
آخَر Fc Ac Ce Ep Ay Ba Bv Bq Fb Fb Fb Bq Bq Bv Ae Ae Ae Ex Ae Ae Ae Av Ap Cm Ej Ej Cu Cu Cu Fe Cu Fe Fe
آخَر Fe Fe Aa Fe Fe Fe Fc Ac Ab Ab Ab Ab Eg Ai Fe Fe Fe Fe Fe Ff Aj Ac Ap Fb Cd Cd Cd El El Bq Ab Ac Ac
آخَر Fl Bn Dz Ez Fb Eo Eo Ay Ay Ep Ep Ep Ax Ep Ay Ay Ep Ep Ay Ep Ax Ay Ep Dc Ax Ay Ay Ep Ay Ay Ay Ay Ay
آخَرَ Dz Ez Fb Eo Eo Ay Ay Ep Ep Ep Ax Ep Ay Ay Ep Ep Ay Ep Ax Ay Ep Dc Ax Ay Ay Ep Ay Ay Ay Ay Ay Ay Ay
آخَرُونَ Bq Fb Fb Fb Bq Bq Bv Ae Ae Ae Ex Ae Ae Ae Av Ap Cm Ej Ej Cu Cu Cu Fe Cu Fe Fe Aa Fe Fe Fe Fc Ac A
آخَرُونَ Ce Ep Ay Ba Bv Bq Fb Fb Fb Bq Bq Bv Ae Ae Ae Ex Ae Ae Ae Av Ap Cm Ej Ej Cu Cu Cu Fe Cu Fe Fe Aa F
آخَرِينَ Eh Ee Ee Bl Bl Eh Cs Eh Eh Eh Bl Ee Ee Fn Fn Dm Bl Fl Aj Eh Ff As Ej Ej As Cm Eg Ap Ab Ab El Cd C
آخَرِينَ Ae Ae Ae Ex Ae Ae Ae Av Ap Cm Ej Ej Cu Cu Cu Fe Cu Fe Fe Aa Fe Fe Fe Fc Ac Ab Ab Ab Ab Eg Ai Fe F
آخِذَةٌ Db Db Ax Fj Fj Ck Fj Fj Ck Ck Fj Ck Ck Ck Ck Ba Ba Bt Bt Bt Ay Ay Ay Dc Ax Ax Ep El Ac Be Aa Aa F
آخِرَ Aj Fl Dz Br Bv Ae Fj Fj Bv Br Fl Eh Eh Eh Eh Eh Eh Eh Eh Eh Fl Bq Eo Bt Bt Ck Av Aq Ck Bv Dz Cs Fh
آخِر Ay Ep Ep Ay Ep Ax Ay Ep Dc Ax Ay Ay Ep Ay Ay Ay Ay Ay Ay Ay Ay Ay Ay Ay Ay Ep Ay Ay Ba Bv Bl Dm Dm
آذَار Ab Ap Eg Ej Fe Fe Fe Fe Fe Fe Ej Cm Eg Aj Aj Aj Eh Ee Ee Bl Bl Eh Cs Eh Eh Eh Bl Ee Ee Fn Fn Dm B
آذَار Af Ef Ef Fi Fi Fi Fi Cx Cu Cm Fl Fl Fl Eg Eg Fe Fa Fa Cp Ei Dn Bc Et Et Et Et Et Ec Ec Ex Av Br B
```

Figure 8-6: Parts of The Dictionary Labeled by Sub-Word Units with Repetitions

The steps in Figure 8-4 were previously explained except the normalization step. The normalization step includes the following:

1- Remove duplicated entries from the dictionary.
2- Remove numbering from numbered pronunciations.
3- Sort the dictionary in ascending order after adding the “sil” entry in different pronunciations like:
    a. sil [ ] sil , which means silent.
    b. <s> [ ] sil, which means <s> (Utterance Start Marker), is always means silent.
    c. </s> [ ] sil , which means </s> (Utterance End Marker) is always means silent.
    d. Silence [ ] sil , the word silence in HTK always means silent.
4- Removing repeated consecutive SWUs and replaces them by only One SWU for the whole dictionary. The result was our final target dictionary. A MATLAB code for automatically generating the dictionary was written and executed.

8.9 Extracting Probabilities and Bigrams for Sub-Word Units

In order to apply hybrid LVQ/HMM model on the extracted SWUs, we were in need of some extra information like: Transition Probability Matrix (TPM) which represents the sub-word unit’s bigrams. TPM entry (a_{ij}) is defined by the probability of going from SWU_i to SWU_j and formally written as:

$$a_{ij} = p(j|i) \tag{8-1}$$

This probability represents the sub-word bigrams model based on the definition of the bigram model in (Rabiner, 1989). Moreover, we were in need for the probability of SWU which represents the probability of the SWU in the corpus and formally defined as:

$$P_{SWU\ i} = \frac{Number\ of\ Time\ SWU\ (i)\ Appears}{Total\ Number\ Of\ Frames} \tag{8-2}$$

For this purpose a MATLAB code is written to extract the probability of each SWU, the transition probability between SWUs and the maximum and minimum of SWU duration (length). The results of this program were passed to the Viterbi algorithm to assist in SWUs recognition.

8.10 Generating Emitting Probability Matrices for SWU

The emitting probability matrix represents the probability of observation (O) given a SWU. Following the same steps of section 7.3 we generated the Emitting probability matrix. The SWU was used in place of the phoneme where, the emitting probability is extracted using the equations 7-1 and 7-2.

8.11 Combining LVQ/HMM Using Viterbi Algorithm for SWUs

The steps in this section are the same as the steps in section 7.3 but, instead of using Arabic phonemes we used the extracted SWUs in the combination. The combination in this way was not adequate for use with the SWUs due to the phoneme duration phenomena. We decided to combine the LVQ with HMM through the HTK tool kit through recognition.

8.12 Chapter Summary

In this chapter we presented the speech segmentation method that we have used to segment speech utterances. We also focused on formulating and generating feature vector matrix. Then, we apply the K-means clustering algorithm on the feature vector matrix to determine a set of centroids that represented the SWUs. We used the K-means algorithm to generate the centroids and label utterances and we used LVQ to improve K-means labeling similar to forced alignment process in Sphinx ASR system. We generated the dictionary based on the SWUs for further use in combining between LVQ and HMM using HTK. Moreover, we concluded that the model used in combining the LVQ and HMM using Viterbi Algorithm through the use of the bigrams and emitting probability extracted through the LVQ is not adequate due to state duration problem.

CHAPTER 9

SPEECH RECOGNITION USING EXTRACTED SUB-WORD UNITS

9.1 INTRODUCTION

As we mentioned earlier in CHAPTER 8, the hybrid LVQ/HMM model was inadequate for transcripting SWUs due to state duration problem. In this chapter we perform two main steps: first we used the LVQ algorithm in transcripting our extracted SWUs. Second, we used those SWUs in Arabic speech recognition through HTK. By applying the second step, it represent the LVQ/HMM combination on the SWU-level which was inadequate when using single-state frame based recognition based on VA.

This chapter is organized as follows: section 9.2 explain the use of the LVQ algorithm in SWUs transcription, section 9.3 investigates the Arabic speech recognition through SWUs where the LVQ/HMM model is embedded in the HTK ASR system, section 9.4 validates the experimental work and discusses the accuracy.

9.2 Arabic Utterance Transcription Using SWUs

We have used the LVQ for transcribing utterances based on SWUs (See section 6.8). We applied the same steps used in section 6.8 and for distance calculation we based upon the flowchart of Figure 6-7. No phonemes bigram models and no aided learning algorithms were used in the transcription. The transcription steps based on LVQ were as follows:

1. Reading the MFCC file of an utterance.
2. Apply the LVQ SWUs alignment algorithm as in Figure 6-7
3. Compare the LVQ transcription of the utterance with its original transcription and calculate the accuracy. The original transcription is the transcription given by K-means algorithm based on the minimum Euclidean distance and improved by LVQ.
4. Repeat steps from 1 to 4 for all testing utterances.
5. Report the average accuracy.

A sensitivity analysis was carried out to investigate the relationship between the number of trained centers and the accuracy of the SWUs transcription. Table 9-1 shows the final accuracy after applying the LVQ on different centers. We tried the centers from 30 to 150. The reason behind choosing 30 as the minimum number of centers is that, we only have 29 letters in Arabic language and we assume that at least we have centers to cover all uttered letters. We choose a maximum of 150 centers due to memory limitations and long time running. Moreover, each Arabic phoneme takes 3-HMM states and we have 50 Arabic phonemes declared by phoneticians, if we multiply 50 by 3, the result would be 150. The graph in Figure 9-1 shows the relationship between the number of centers and accuracy. We did not found big difference between the accuracy when we used 90 or 150 centers. This means, increasing the number of centers (SWUs) did not mean increasing the SWUs transcription accuracy. The reason behind this is that, the

SWUs accuracy will either increased or stay steady for sometimes then, it starts falling down again, which means that the number of centers did not represent the speech data. This phenomena, ensures the existence of the best or optimal representative number of centers (SWUs). We repeated the transcription by varying the number of centers from 30 up to 150 and the best result was when we used 70 centers.

When we compared our result with the result of using Arabic phonemes, we found that, the phoneme accuracy using standalone LVQ was 72% while it is 79.3% when using LVQ with SWUs.

Table 9-1: Relationship between number of centers (SWUs) and SWUs Transcription Accuracy

Number Of Trained Centers	Average SWU Transcription Accuracy
30	40%
50	75%
70	79.3%
90	76.2%
150	75.8%

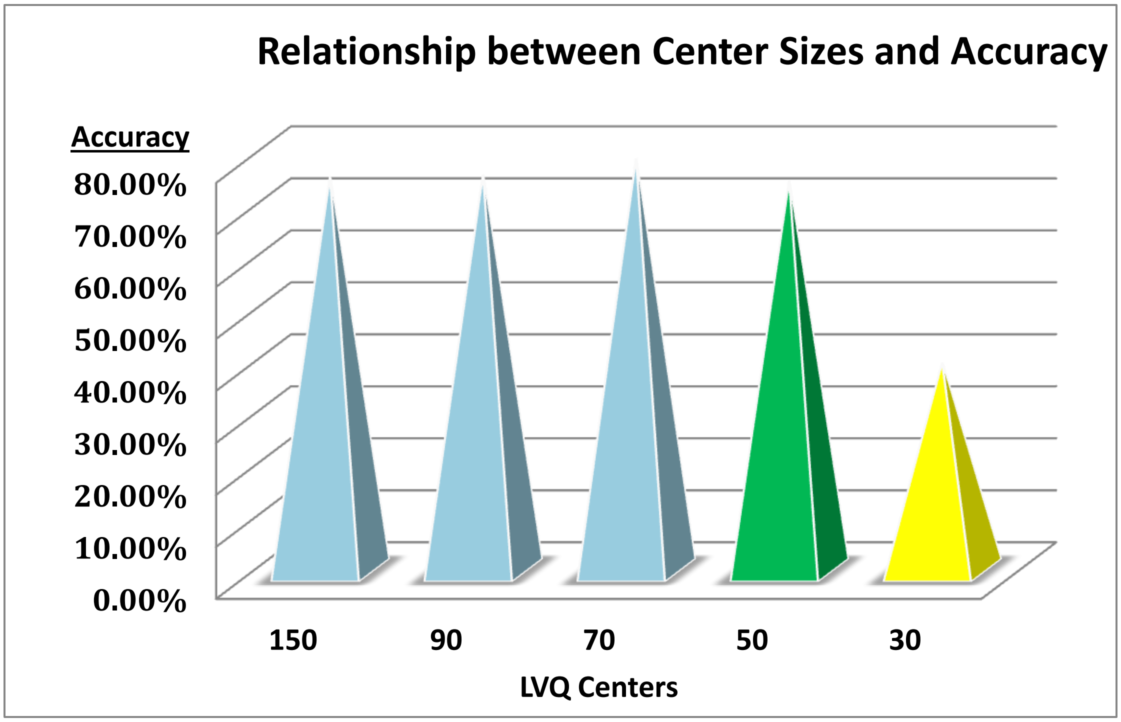


Figure 9-1: Number of Centers and Accuracy Relationship

Using SWUs in Arabic Speech Recognition (LVQ/HMM Model)

In chapter 8, we have reached an acceptable labeling using K-means. Then we improved this labeling using LVQ. Based on this latest labeling we generated the dictionary. In this section we used the extracted SWUs based on K-means/LVQ learning algorithms in continuous Arabic speech recognition. The steps for doing the recognition are as follows:

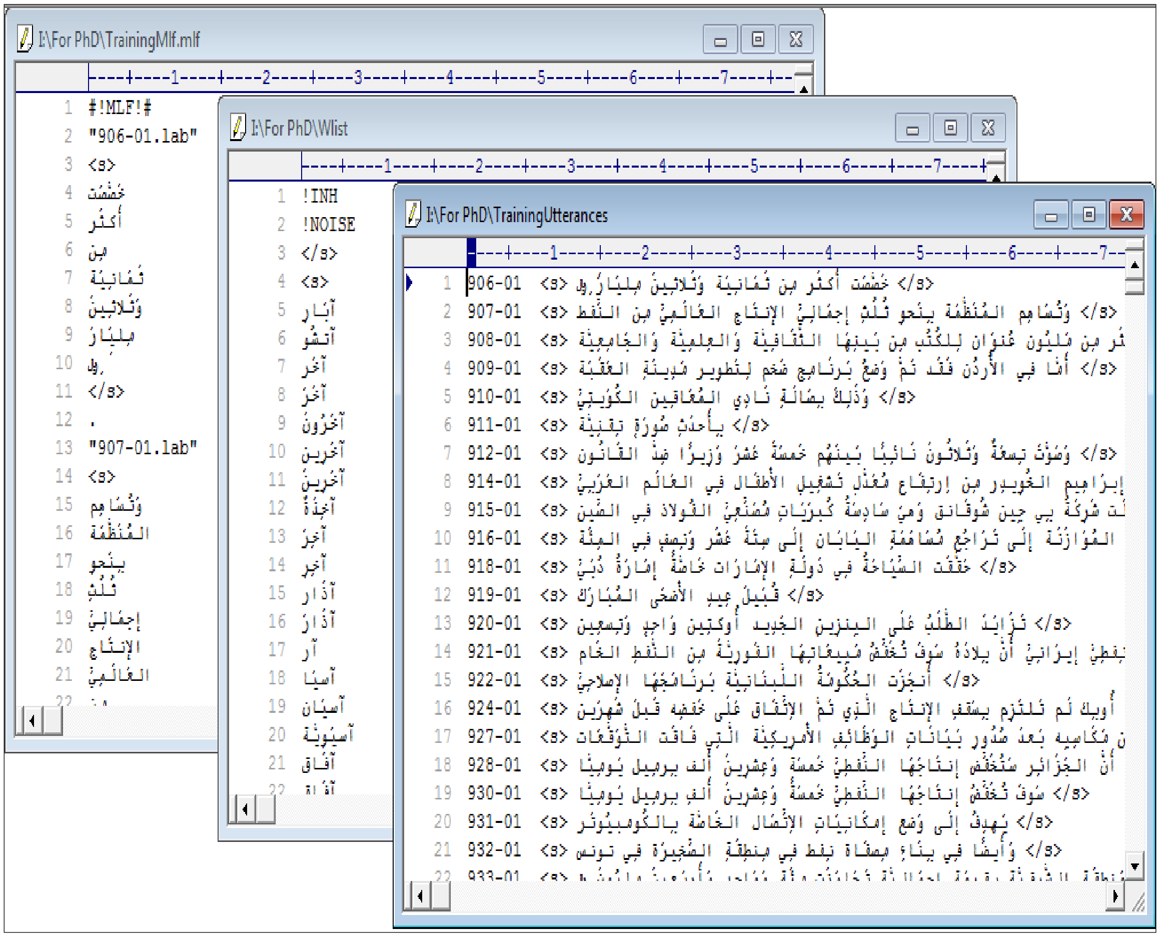
1. Generating the wordlist from the dictionary: this could be done either manually or using the Perl command “prompt2word”.

The “prompt2word” is a Perl-language code that already comes with sphinx ASR-system and HTK-system; it converts the transcription file to word list transcription. Figure 9-2 shows the corpus utterances in front window and the generated word list in the middle window.

2. Generating word level master label file (MLF) file from the utterances text that we already have in the corpus, a Perl-language code similar to “prompt2word” is used, it is “prompt2mlf” Perl Command.

The command takes the transcription of the utterances and produces the words of each utterance with a header of its number for further use. Figure 9-2 shows the MLF file at the rear window.

3. Generating SWUs level transcription (MLF) file by the HTK command HLed. The HLed command is an editing command which has its own script language to manipulate the dictionary and the word-level MLF file. It produces the SWUs level master label file (SWUs MLF file). Figure 9-3 shows the process, where the script files that manipulates the dictionary is fed to HLED along with word-level transcription to produce SWUs-level transcription. The script file command EX means: replace every word in the utterance with its transcription from the dictionary. The command IS means: insert the silent model sil at the start and end of each utterance. The command DE means; delete the short pause spaces (Young et al., 2006).

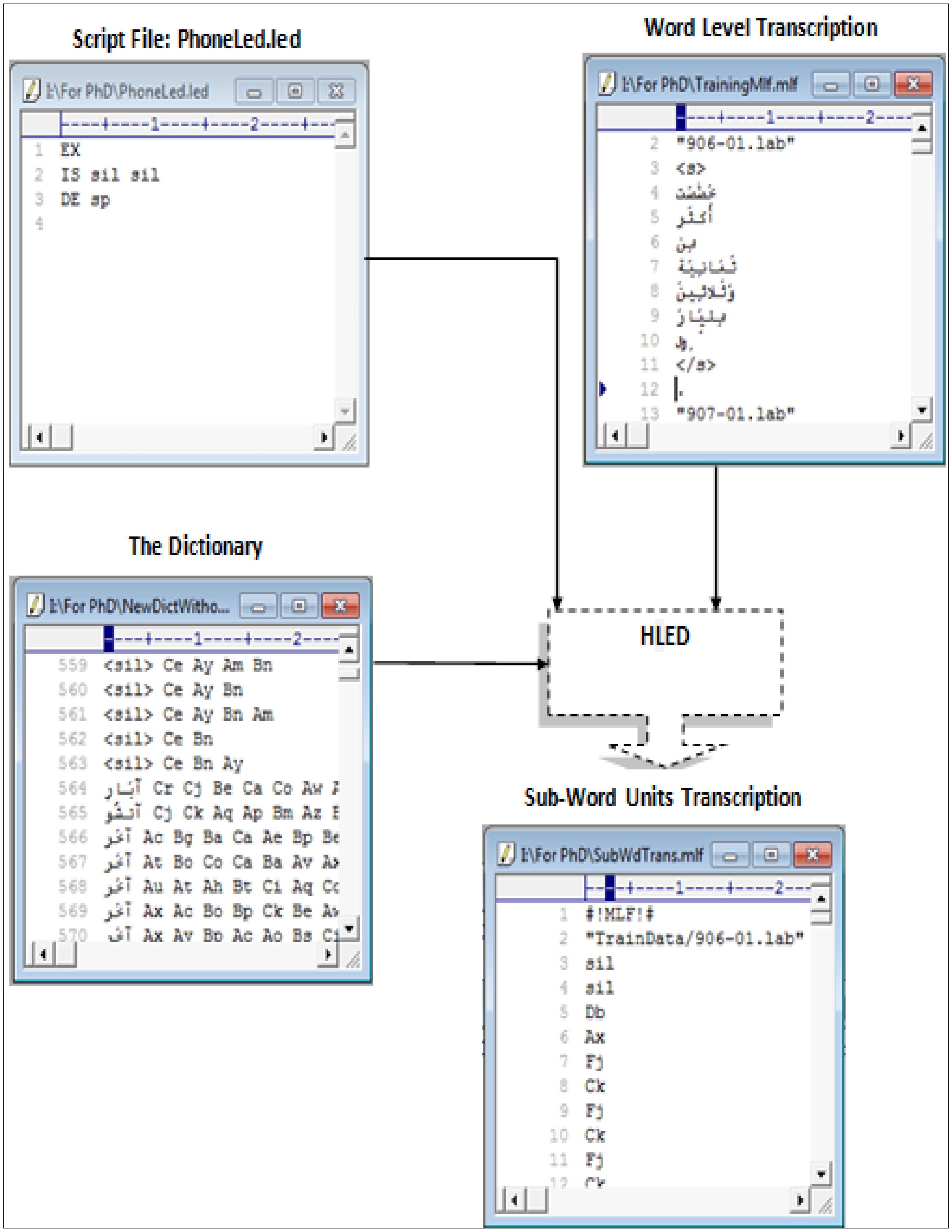


```
#!MLF!#
"906-01.lab"
<s>
خَفَضَت
أَكثَر
مِن
ثَمانِيَة
وَثَلاثِينَ
مِليار
رِيال
</s>
.
"907-01.lab"
<s>
وَتُساهِم
المُنَظَّمَة
بِنَحو
ثُلُث
إجمالِيّ
الإنتاج
العالَمِيّ
```

```
!INH
!NOISE
</s>
<s>
آبار
آتشو
آخَر
آخَر
آخَرُون
آخَرِين
آخَرِينَ
آخِذَة
آخِر
آخِر
آذار
آذار
آر
آسيا
آسيان
آسيَوِيَّة
آفاق
```

```
906-01 <s> خَفَضَت أَكثَر مِن ثَمانِيَة وَثَلاثِينَ مِليار رِيال </s>
907-01 <s> وَتُساهِم المُنَظَّمَة بِنَحو ثُلُث إجمالِيّ الإنتاج العالَمِيّ مِن النَّفط </s>
908-01 <s> ثَر مِن مِليُون عُنوان لِلكُتُب مِن بَينِها الثَّقافِيَّة وَالعِلمِيَّة وَالجامِعِيَّة
909-01 <s> أَمّا فِي الأُردُن فَقَد تَمَّ وَضع بَرنامِج ضَخم لِتَطوِير مَدِينَة العَقَبَة </s>
910-01 <s> وَذَلِكَ بِضِيافَة نادِي المُعاقِين الكُوَيتِيّ </s>
911-01 <s> بِأَحدَث صُورَة تِقنِيَّة </s>
912-01 <s> وَصَوَّتَ تِسعَة وَثَلاثُونَ نائِبًا بَينَهُم خَمسَة عَشَر وَزِيرًا ضِدّ القانُون </s>
914-01 <s> إبراهِيم الغُوَيدِر مِن اِرتِفاع مُعَدَّل تَشغِيل الأَطفال فِي العالَم العَرَبِيّ
915-01 <s> ـت شَرِكَة بِي جِين شُوقانق وَهِيَ سادِسَة كُبرَيات مُصَنِّعِي الفُولاذ فِي الصِّين
916-01 <s> المُوازَنَة إِلَى تَراجُع مُساهَمَة اليابان إِلَى سِتَّة عَشَر وَنِصف فِي المِئَة
918-01 <s> حَقَّقَت السِّياحَة فِي دَولَة الإمارات خاصَّة إمارَة دُبَيّ </s>
919-01 <s> قُبَيلَ عِيد الأَضحَى المُبارَك </s>
920-01 <s> تَزايَدَ الطَّلَب عَلَى البِنزِين الجَدِيد أُوكتِين واحِد وَتِسعِين </s>
921-01 <s> نَفطِيّ إيرانِيّ أَنَّ بِلادَهُ سَوفَ تُخَفِّض مَبِيعاتها الفَورِيَّة مِن النَّفط الخام
922-01 <s> أَنجَزَت الحُكُومَة اللُّبنانِيَّة بَرنامَجَها الإصلاحِيّ </s>
924-01 <s> أُوبِك لَم تَلتَزِم بِسَقف الإنتاج الَّذِي تَمَّ الاتِّفاق عَلَى خَفضِه قَبلَ شَهرَين
927-01 <s> ن مَكاسِبِه بَعدَ صُدُور بَيانات الوَظائِف الأَمرِيكِيَّة الَّتِي فاقَت التَّوَقُّعات
928-01 <s> أَنَّ الجَزائِر سَتُخَفِّض إنتاجَها النَّفطِيّ خَمسَة وَعِشرِينَ أَلف بِرمِيل يَومِيًّا
930-01 <s> سَوفَ تُخَفِّض إنتاجَها النَّفطِيّ خَمسَة وَعِشرِينَ أَلف بِرمِيل يَومِيًّا </s>
931-01 <s> يَهدِف إِلَى وَضع إمكانِيّات الاتِّصال الخاصَّة بِالكُمبيُوتَر </s>
932-01 <s> وَأَيضًا فِي بِناء مِصفاة نِفط فِي مِنطَقَة الصُّخَيرَة فِي تُونِس </s>
```

Figure 9-2: Utterances (Front Window), Word List (Middle Window) and Utterance MLF file (Rear Window).


Script File: PhoneLed.led
I:\For PhD\PhoneLed.led
Word Level Transcription
I:\For PhD\TrainingMlf.mlf
The Dictionary
I:\For PhD\NewDictWitho...
HLED
Sub-Word Units Transcription
I:\For PhD\SubWdTrans.mlf

```
EX
IS sil sil
DE sp
```

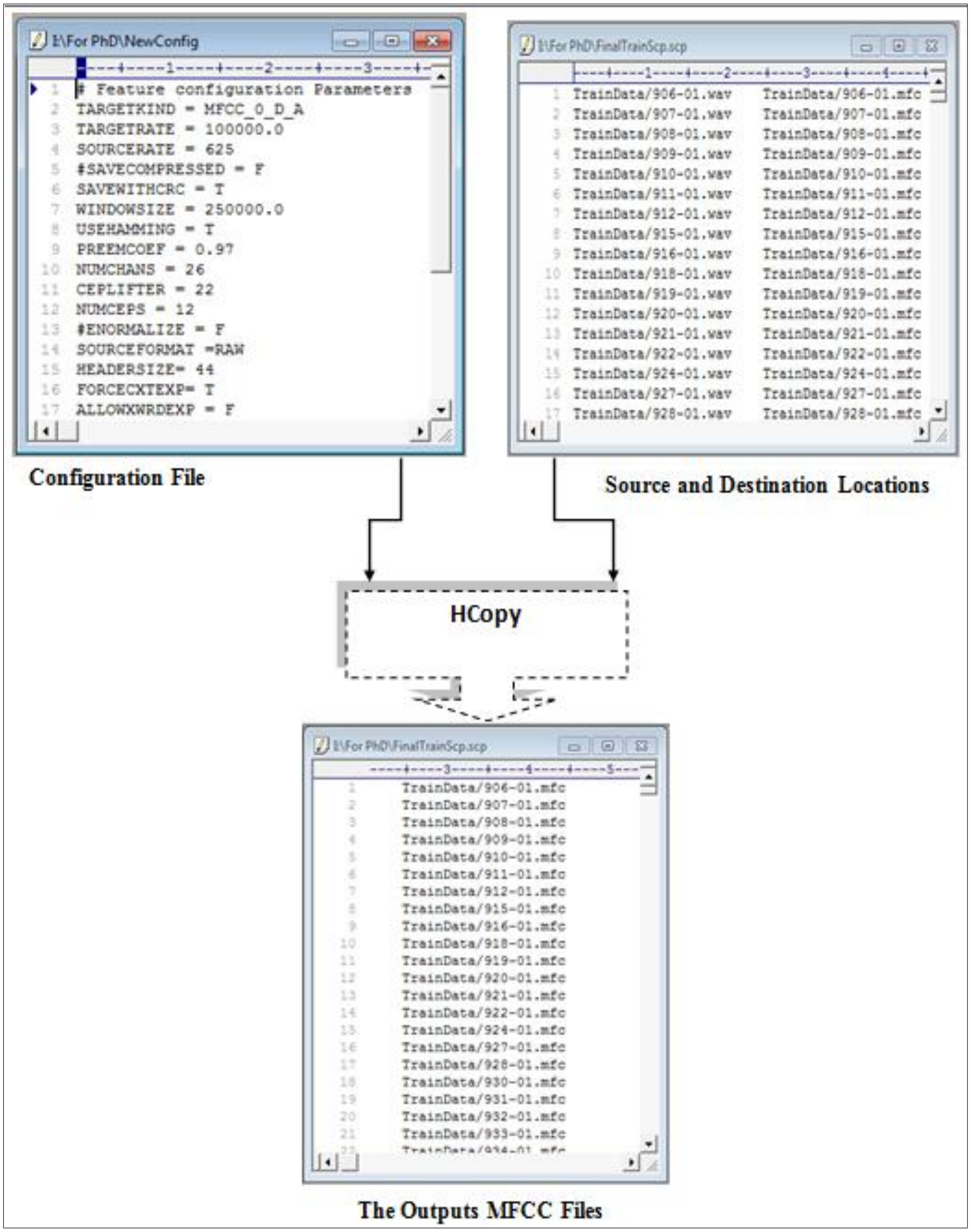
```
"906-01.lab"
<s>
تنفق
أكثر
من
ثمانية
وثلاثين
بليار
ه.
</s>
.
"907-01.lab"
```

```
<sil> Ce Ay Am Bn
<sil> Ce Ay Bn
<sil> Ce Ay Bn Am
<sil> Ce Bn
<sil> Ce Bn Ay
آبار Cr Cj Be Ca Co Aw
آنشو Cj Ck Aq Ap Bm Az
آخر Ac Bg Ba Ca Ae Bp
آخر At Bo Co Ca Ba Av
آخر Au At Ah Bt Ci Aq
آخر Ax Ac Bo Bp Ck Be
آف Ax Av Bo Ac Ao Bs
```

```
#!MLF!#
"TrainData/906-01.lab"
sil
sil
Db
Ax
Fj
Ck
Fj
Ck
Fj
```

Figure 9-3: Generating SWU Level Transcription

4. Converting the WAVE files into their suitable MFCC files, so that they are recognized and accepted by the HTK-tool. A configuration setup file is generated based on the properties of the WAVE file specifications. The HTK-HCopy command was used to do the conversion after providing it with all necessary files. The configuration that we used and the source WAVE file location with the target MFCC file location as a script file are fed to HCopy command. Figure 9-4 shows the required files for conversion. We already converted the wave files in the corpus and set the configuration file based on their attributes. The details of the configuration parameters could be found in (Young et al., 2006).



```
# Feature configuration Parameters
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SOURCERATE = 625
#SAVECOMPRESSED = F
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
#ENORMALIZE = F
SOURCEFORMAT =RAW
HEADERSIZE= 44
FORCECXTEXP= T
ALLOWXWRDEXP = F
```

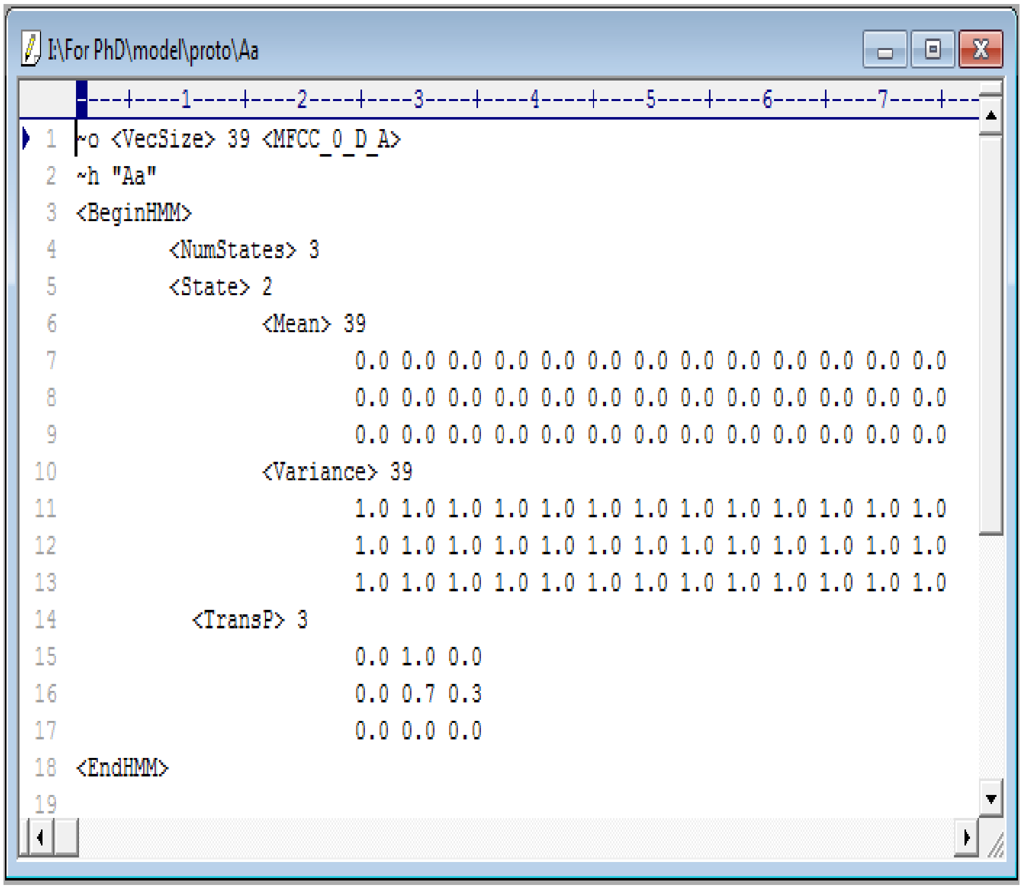
```
TrainData/906-01.wav    TrainData/906-01.mfc
TrainData/907-01.wav    TrainData/907-01.mfc
TrainData/908-01.wav    TrainData/908-01.mfc
TrainData/909-01.wav    TrainData/909-01.mfc
TrainData/910-01.wav    TrainData/910-01.mfc
TrainData/911-01.wav    TrainData/911-01.mfc
TrainData/912-01.wav    TrainData/912-01.mfc
TrainData/915-01.wav    TrainData/915-01.mfc
TrainData/916-01.wav    TrainData/916-01.mfc
TrainData/918-01.wav    TrainData/918-01.mfc
TrainData/919-01.wav    TrainData/919-01.mfc
TrainData/920-01.wav    TrainData/920-01.mfc
TrainData/921-01.wav    TrainData/921-01.mfc
TrainData/922-01.wav    TrainData/922-01.mfc
TrainData/924-01.wav    TrainData/924-01.mfc
TrainData/927-01.wav    TrainData/927-01.mfc
TrainData/928-01.wav    TrainData/928-01.mfc
```

Source and Destination Locations
HCopy
I:\For PhD\FinalTrainScp.scp

```
TrainData/906-01.mfc
TrainData/907-01.mfc
TrainData/908-01.mfc
TrainData/909-01.mfc
TrainData/910-01.mfc
TrainData/911-01.mfc
TrainData/912-01.mfc
TrainData/915-01.mfc
TrainData/916-01.mfc
TrainData/918-01.mfc
TrainData/919-01.mfc
TrainData/920-01.mfc
TrainData/921-01.mfc
TrainData/922-01.mfc
TrainData/924-01.mfc
TrainData/927-01.mfc
TrainData/928-01.mfc
TrainData/930-01.mfc
TrainData/931-01.mfc
TrainData/932-01.mfc
TrainData/933-01.mfc
TrainData/934-01.mfc
```

Figure 9-4: Converting WAVE Files to MFCC Files

5. Preparing and initializing the HMM file: in our case we used the single emitting state HMM model. HTK defines the single emitting state HMM by 3-states: the first one is non-emitting state for start of SWU, the emitting state at the middle and non-emitting state at the end of SWU. Figure 9-5 shows the prototype of HMM definition file. In HTK we have two types of initializations for the HMM: Normal initialization where the boundary of the SWUs is already known, and the flat initialization where we don't know the boundaries of SWUs. The flat initialization is done based on the flowchart in Figure 9-6. We provided the initialization command HCompV with the HMM's prototypes and with transcription of the utterances. HCompV recalculated the mean and variance of each HMM state and stored the result in a new directory. The command is executed for each SWU. the details of the command and its required parameters and switches could be found in (Young et al., 2006).



```
~o <VecSize> 39 <MFCC_0_D_A>
~h "Aa"
<BeginHMM>
        <NumStates> 3
        <State> 2
                <Mean> 39
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                        0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                <Variance> 39
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
                        1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
          <TransP> 3
                        0.0 1.0 0.0
                        0.0 0.7 0.3
                        0.0 0.0 0.0
<EndHMM>
```

Figure 9-5: Single emitting State HMM for 'Aa' SWU

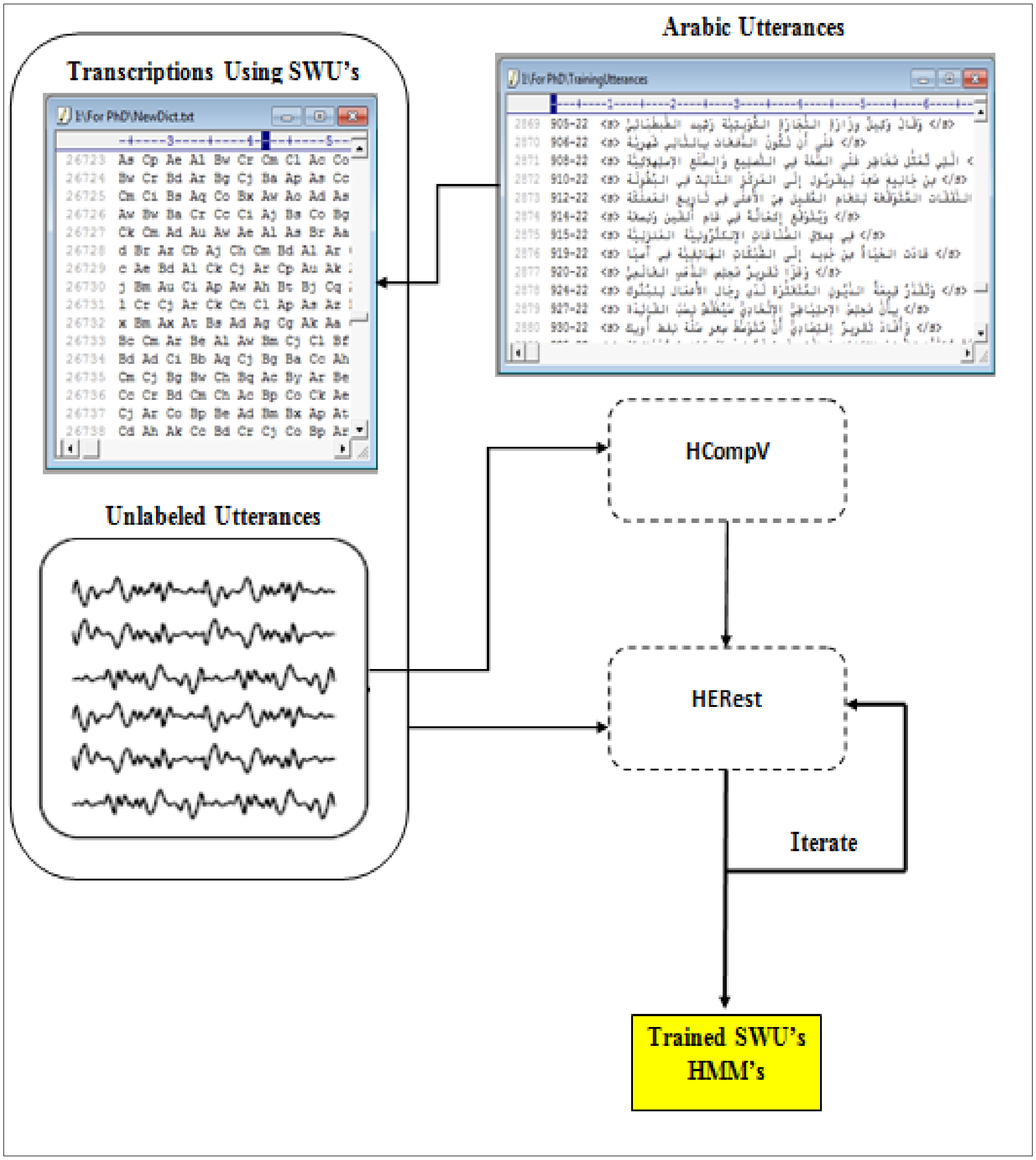
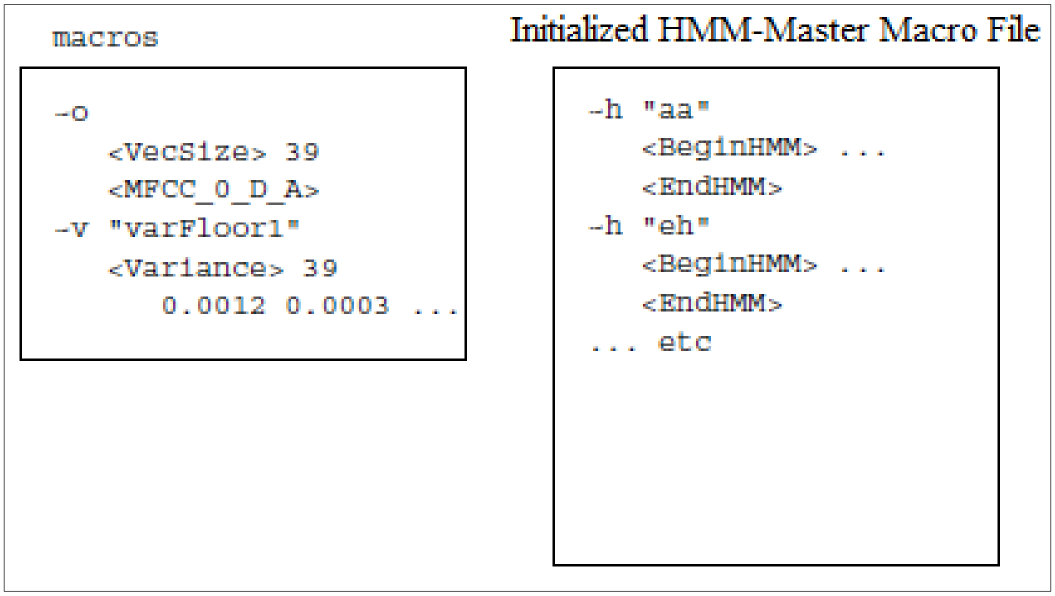


Figure 9-6: Initializing SWUs HMM

6. Romanizing Arabic transcription since the HTK does not support Arabic code. For this purpose a MATLAB code was written to do the Romanization automatically. All utterances must be romanized including all files needed for recognition which are written in Arabic.

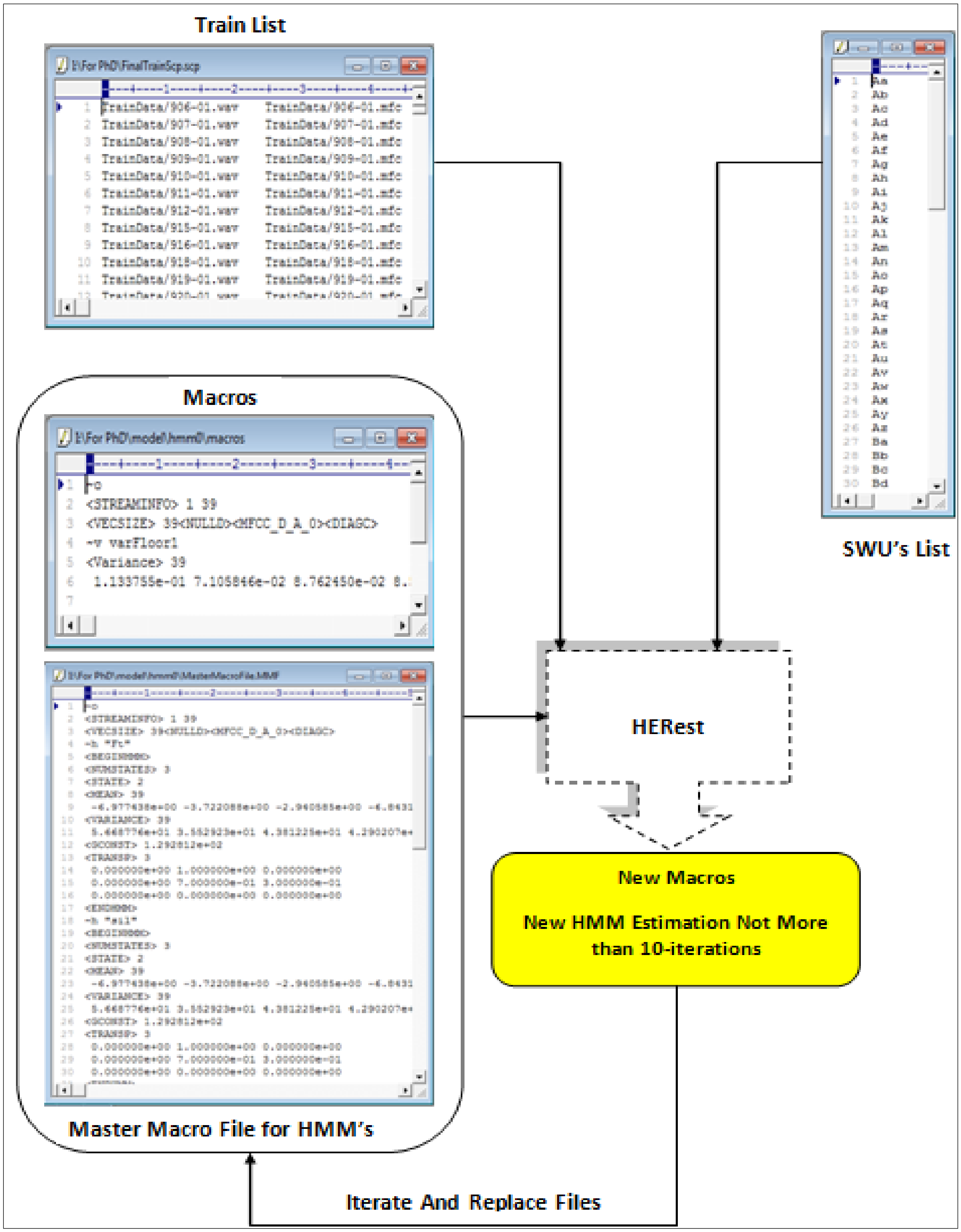
7. Training HMM's using Baum-Welch algorithm represented by the HERest command in HTK. For this purpose another two important files are created: the macro file which contains the variance floor and the master macro file which contains all initialized HMM's in one file. Figure 9-7 shows both the macro file and the master macro file. The training process needs those files in addition to other necessary files. The whole training process and its necessary files are shown in Figure 9-8.



```
~o
   <VecSize> 39
   <MFCC_0_D_A>
~v "varFloor1"
   <Variance> 39
      0.0012 0.0003 ...
```

```
~h "aa"
   <BeginHMM> ...
   <EndHMM>
~h "eh"
   <BeginHMM> ...
   <EndHMM>
... etc
```

Figure 9-7: Macros and Master Macro Files (Young et al., 2006)


Train List
I:\For PhD\FinalTrainScp.scp
Macros
I:\For PhD\model\hmm0\macros
I:\For PhD\model\hmm0\MasterMacroFile.MMF
Master Macro File for HMM's
SWU's List
HERest
New Macros
New HMM Estimation Not More than 10-iterations
Iterate And Replace Files

```
TrainData/906-01.wav     TrainData/906-01.mfc
TrainData/907-01.wav     TrainData/907-01.mfc
TrainData/908-01.wav     TrainData/908-01.mfc
TrainData/909-01.wav     TrainData/909-01.mfc
TrainData/910-01.wav     TrainData/910-01.mfc
TrainData/911-01.wav     TrainData/911-01.mfc
TrainData/912-01.wav     TrainData/912-01.mfc
TrainData/915-01.wav     TrainData/915-01.mfc
TrainData/916-01.wav     TrainData/916-01.mfc
TrainData/918-01.wav     TrainData/918-01.mfc
TrainData/919-01.wav     TrainData/919-01.mfc
TrainData/920-01.wav     TrainData/920-01.mfc
```

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
~v varFloor1
<Variance> 39
 1.133755e-01 7.105846e-02 8.762450e-02 8.
```

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
~h "Fc"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 39
 -6.977438e+00 -3.722088e+00 -2.940585e+00 -6.8431
<VARIANCE> 39
 5.668776e+01 3.552923e+01 4.381225e+01 4.290207e+
<GCONST> 1.292812e+02
<TRANSP> 3
 0.000000e+00 1.000000e+00 0.000000e+00
 0.000000e+00 7.000000e-01 3.000000e-01
 0.000000e+00 0.000000e+00 0.000000e+00
<ENDHMM>
~h "ai1"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 39
 -6.977438e+00 -3.722088e+00 -2.940585e+00 -6.8431
<VARIANCE> 39
 5.668776e+01 3.552923e+01 4.381225e+01 4.290207e+
<GCONST> 1.292812e+02
<TRANSP> 3
 0.000000e+00 1.000000e+00 0.000000e+00
 0.000000e+00 7.000000e-01 3.000000e-01
 0.000000e+00 0.000000e+00 0.000000e+00
```

```
Aa
Ab
Ac
Ad
Ae
Af
Ag
Ah
Ai
Aj
Ak
Al
Am
An
Ao
Ap
Aq
Ar
As
At
Au
Av
Aw
Ax
Ay
Az
Ba
Bb
Bc
Bd
```

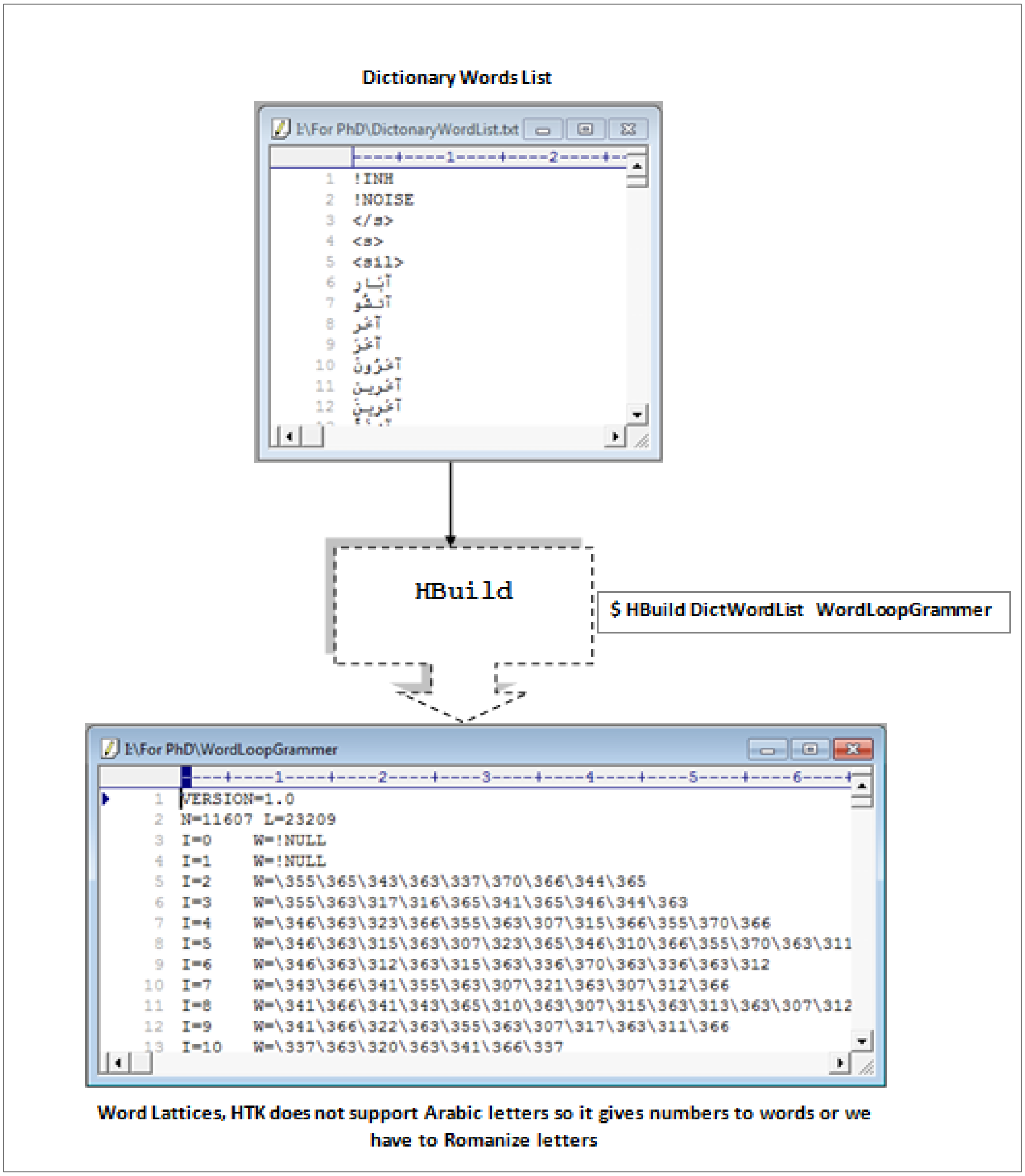
Figure 9-8: The Training Process

Building the word network lattices, where we do not have any grammar. The only grammar that we have is the words followed by each other (loops of words). We built the words network using the command HBuild from the HTK. The commands format, input and output are shown in

8. Figure 9-9 where the words follow each other, the word network appear with numbers of 4-digits representing Arabic words since the HTK can't identify Arabic letters.

9. Using Viterbi algorithm through HVite HTK-command the best alignment for a testing set of utterances was derived. The recognition summary was reported through HResult HTK-command. The HResult command compares the recognition result from HVite command with a referenced transcription file and calculated accuracy, correctness, WER. Through this command we can extract the confusion matrix and many other statistics.

10. Figure 9-10 shows the flow of HVite and HResult commands and there necessary files. Section 9.3 shows the result of our experiments.



```
!INH
!NOISE
</s>
<s>
<sil>
آبـار
آنـفو
آخر
آخز
آخَرُون
آخَرِين
آخَرِينَ
```

```
$ HBuild DictWordList  WordLoopGrammer
```

```
VERSION=1.0
N=11607 L=23209
I=0     W=!NULL
I=1     W=!NULL
I=2     W=\355\365\343\363\337\370\366\344\365
I=3     W=\355\363\317\316\365\341\365\346\344\363
I=4     W=\346\363\323\366\355\363\307\315\366\355\370\366
I=5     W=\346\363\315\363\307\323\365\346\310\366\355\370\363\311
I=6     W=\346\363\312\363\315\363\336\370\363\336\363\312
I=7     W=\343\366\341\355\363\307\321\363\307\312\366
I=8     W=\341\366\341\343\365\310\363\307\315\363\313\363\307\312
I=9     W=\341\366\322\363\355\363\307\317\363\311\366
I=10    W=\337\363\320\363\341\366\337
```

Word Lattices, HTK does not support Arabic letters so it gives numbers to words or we have to Romanize letters

Figure 9-9: Building Word Grammar Loop

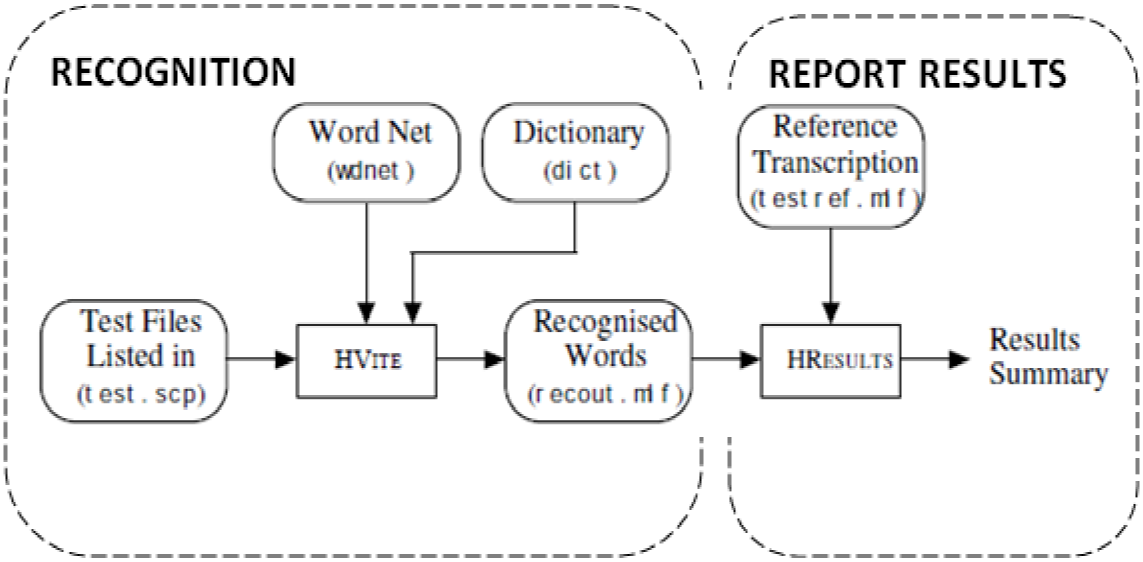


Figure 9-10: Recognition through Viterbi Algorithm (Young et al., 2006)

9.3 Validation and Accuracy Measurement

Multiple recognition experiments were carried out using different numbers of SWUs (Centers). Table 9-2 presents the accuracy of recognition.

The results on the word level were not encouraging and need further investigation. Since it is the first time that we use the Arabic SWUs in speech recognition and this was rarely tackled, we did not expect much accuracy at the beginning. As a result we have to investigate the possible sources of errors that may cause a significant recognition rate reduction like:

1- The labeling accuracy.

2- Number of HMM’s used (one in our case).

3- MFCC representation in HTK when converting from wave to MFCC.

In Figure 9-11 and when transcribing SWUs, The 70 centers gave the best results in transcription, and they gave the best word level recognition rate also. The recognition rate is going down after 70 centers. That means, when using one-state HMM, the 70 centers was the best size that we obtained the best recognition rate when using them. Unfortunately, the recognition rate was low and need more investigation as we mentioned earlier. Correctness and accuracy are calculated by HTK using the equations 5-1 and 5-2 respectively.

9.4 Chapter Summary

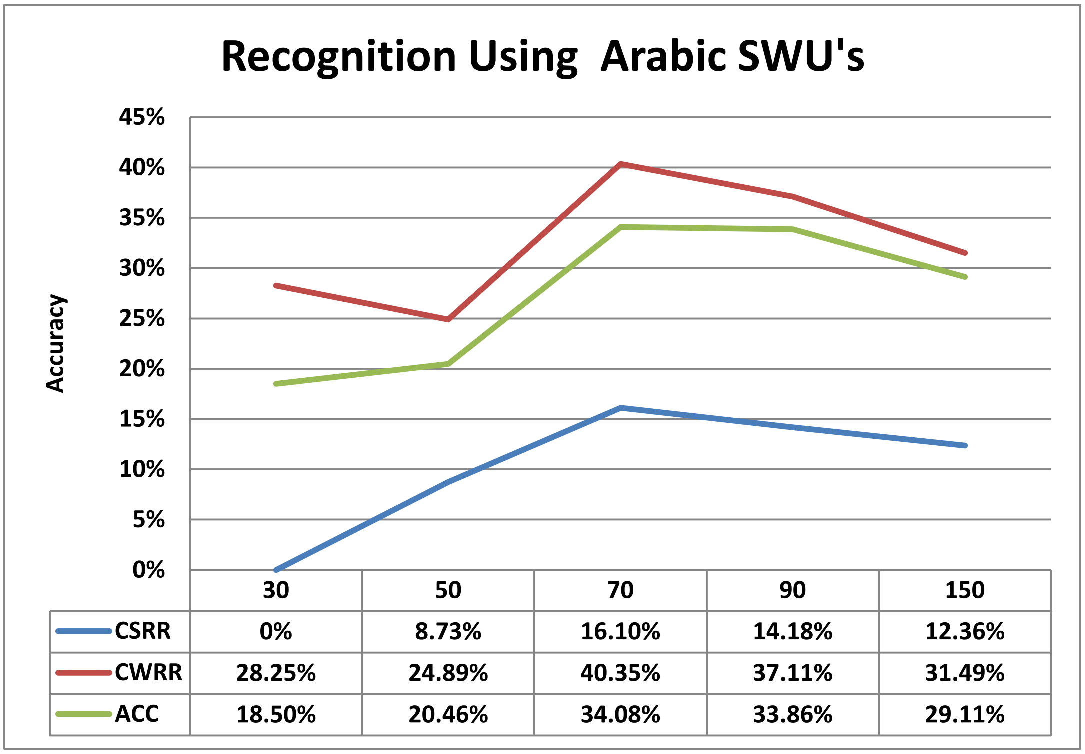
In this chapter, we checked the continuous Arabic speech recognition using extracted SWUs. We found that, the behavior of recognition at the SWUs level is the same at the word level except the recognition rate. At the SWUs transcription level the best rate was

79.3% which considered being high and good and this rate was obtained on the size of 70 centers. At the word level transcription done using the HTK tool-kit, the best recognition rate was low (34.08%) but, it also obtained when using the size of 70 centers.

We believe that the word-level recognition rate is much higher than this ratio. Our research in this dissertation limited to SWUs-level and what we have done on the word-level is just an initial indicator for future work to be done on this level. As we mentioned before, multiple sources of errors affects the word level recognition and need to be deeply investigated.

Table 9-2: Results for different SWU Centers

Centers	Correct Sentence Recognition Rate	Correct Word Recognition Rate	Accuracy
30	0%	28.25%	18.5%
50	8.73%	24.89%	20.46%
70	16.1%	40.35%	34.08%
90	14.18%	37.11%	33.86%
150	12.36%	31.49%	29.11%



	30	50	70	90	150
CSRR	0%	8.73%	16.10%	14.18%	12.36%
CWRR	28.25%	24.89%	40.35%	37.11%	31.49%
ACC	18.50%	20.46%	34.08%	33.86%	29.11%

Figure 9-11: Recognition Results Using Arabic SWUs for Different Centers.

CHAPTER 10

CONCLUSION AND FUTURE WORK

10.1 Conclusion and Summary

Our research work investigated the automatic extraction of Arabic SWUs for Arabic continuous speech recognition. The way of SWUs extraction comprised two main integrated branches: the first branch was the study of the existing Arabic phonemes, and the second branch was the use of the results of the first branch to assist in the extraction of Arabic SWUs.

With respect to SWUs extraction, we have achieved the following:

- We have pursued a comprehensive statistical analysis of the Arabic phonemes to investigate the probability behavior of the phonemes during the speech. For each phoneme we succeeded in: clustering the phonemes based on their lengths, extracting the probability of each one, extracting phoneme bigrams, triphone frequencies, the length distribution, the maximum and the minimum duration of each phoneme. This statistical study was published and used in subsequent investigations see (Nahar et al., 2012).
- We investigated the use of the data driven approach for Arabic phoneme transcription with varying number of HMM states. Our study recommended the number of HMM nonemitting states for each Arabic phoneme. The results achieved better phoneme recognition rates when we

assigned each phoneme its suitable number of HMM nonemitting states. Some phonemes took only one nonemitting state like the phoneme “T” and with excellent recognition.

- We investigated the use of the LVQ without the use of phoneme bigrams and LVQ combination with HMM in Arabic phoneme transcriptions. Using only LVQ model, we achieved 72% of accuracy for phoneme transcriptions.
- In the LVQ/HMM phoneme transcription we studied the problem of state duration control (when to force the HMM to exit from its current state instead of looping). We modified an existing Viterbi algorithm that considers the phoneme duration in its recognition part, in order to use it in our Arabic phoneme transcription. This was achieved by adding the phoneme bigram model to the algorithm and restructuring its steps. The average accuracy achieved was 89% which is a comparable improvement with respect to standalone LVQ where 72% of accuracy was achieved.
- We suggested a number of centers (SWUs) for the K-means algorithm, then we trained them using LVQ until reaching zero classification errors. Then, we labeled utterances based on these centers by considering the order and length of the utterance in the feature vector matrix. Moreover, we labeled words based on their orders and lengths in the utterance itself. We used Sphinx ASR system for words segmentation.
- Using standalone LVQ, we tested the SWUs transcription and, we achieved promising results. Since we used the frame based labeling and

we did not take neighboring correlation into consideration, we discovered that using LVQ/HMM model causes a state duration problem. We decided to perform LVQ/HMM combination using the HTK-tool, the recognition was done based on the SWUs.

- We automatically generated the dictionary based on the new word transcriptions.

We conclude that the K-means and LVQ could be used for extracting the SWUs. The K-means extracted the SWUs and the LVQ trained the representative codebooks of these SWUs and fixed the K-means labeling by forcing the alignment between utterances and new labels. At the SWUs level, LVQ results were promising while LVQ/HMM model suffered from state duration control problem.

We carried out a sensitivity analysis to find the relation between the number of representative centers of the speech features (SWUs) and SWUs transcription accuracy. We experimented with different center sizes (30, 50, 70, 90, and 150) and found that the accuracy dropped after we exceeded the size 70. Based on the Arabic corpus and our experiments we found that the best SWUs inventory size is approximately 70. However, this may be affected by the used corpus.

We used the HTK-tool for recognition based on the extracted SWUs since we have the newly generated dictionary and new SWUs. The accuracy however was low and the word error rate was high. Some frames may have been biased to some centers leading to the relatively low performance.

10.2 Future Work

Several research directions emerged from our research. The following future directions could be considered:

1- Since we used the word segmentation that resulted from the forced alignment produced by Sphinx ASR system, we were able to determine the start and end of each word in the utterances. Whereas, if we assume that we do not have the word segmentation then, a future research needs to be conducted to tackle Word –Boundary detection problem.

2- The state duration problem that appeared when we try to apply the LVQ/HMM model on SWUs needs to be addressed. An investigation could be done to seek for the proper duration function that can give the same or better results compared with Arabic Phonemes results.

3- Multi-state SWUs needs to be examined and compared with our single nonemitting state SWUs.

REFERENCES

Abdul-Kadir, N., Sudirman, R., 2011. Difficulties of standard arabic phonemes spoken by non-arab primary school children based on formant frequencies. Journal of Computer Science 7, 1003–1010.

Abuzeina, D., Al-Khatib, W., Almuhtaseb, H., Elshafei, M., 2011. Cross-word Arabic pronunciation variation modeling for speech recognition. International Journal of Speech Technology 14, 227–236.

AbuZeina, D., Al-Khatib, W., Elshafei, M., Al-Muhtaseb, H., 2011. Within-word pronunciation variation modeling for Arabic ASRs: a direct data-driven approach. International Journal of Speech Technology 15, 65–75.

Adami, A., Hermansky, H., 2003. Segmentation of speech for speaker and language recognition, in: 8th European Conference on Speech Communication and Technology, EUROSPEECH 2003. Geneva, Switzerland, pp. 1–4.

Ahmed, M., 1991. Toward an Arabic text-to-speech system. The Arabian Journal for Science and Engineering 16, 565–583.

Aimetti1, G., Moore, R.K., Ten Bosch, L., 2010. Discovering an Optimal Set of Minimally Contrasting Acoustic Speech Units: A Point of Focus for Whole-Word Pattern Matching, in: Proc. Interspeech,2010. pp. 310–313.

Aksungurlu, T., 2008. "Investigation of Automatically Derived Subword Units for Turkish LVCSR", Master Thesis, Bogazici University.

Alghamdi, M., EL Hadj, Y., Al Kanhal, M., 2007. A Manual System to Segment and Transcribe Arabic Speech, in: IEEE International Conference on Signal Processing and Communication (ICSPC07). Dubai, UAE, pp. 24–27.

Alghamdi, M., Elshafei, M., Almuhtaseb, H., 2009. Arabic broadcast news transcription system. International Journal of Speech Technology 10, 183–195.

Ali, M., Elshafei, M., Alghamdi, M., Al-Muhtaseb, H., Al-Najjar, A., 2008. Generation of Arabic phonetic dictionaries for speech recognition, in: In the 5th International Conference on Innovations in Information Technology, 2008. IIT 2008. United Arab Emarates, pp. 59–63, 16–18.

Ali, M., Elshafei, M., Alghamdi, M., Al-Muhtaseb, H., Al-Najjar, A., 2009. Arabic phonetic dictionaries for speech recognition. Journal of Information Technology Research 2, 67–80.

Ali, M., Elshafei, M., Al-Ghamdi, M., Al-Muhtaseb, H., Al-Najjar, A., 2008. Arabic phonetic dictionaries for speech recognition, in: Innovations in Information Technology, 2008. IIT 2008. International Conference On. IEEE, pp. 59–63.

Al-Manie, M., Alkanhal, M., 2009. Automatic speech segmentation using the Arabic phonetic database, in: Proceedings of the 10th WSEAS International Conference on Automation & Information. Prague, Czech Republic, pp. 76–79.

Al-Manie, M., Alkanhal, M., Al-Ghamdi, M., 2010. Arabic speech segmentation: Automatic verses manual method and zero crossing measurements. Indian Journal of Science and Technology 3, 1134–1138.

Alpaydin, E., 2004. 133-129 Introduction to machine learning. Cambridge, Massachusetts: MIT Press.

Al-Zabibi, M., 1990. 1-266 "An acoustic-phonetic approach in automatic Arabic speech recognition". Loughborough University.

Anwar, M., Awais, M., Masud, S., Shamail, S., 2006. Automatic Arabic Speech Segmentation System. International Journal of information Technology 12, 102–111.

ÄŒernocký, J., Baudoin, G., Petrovska, D., Hennebert, J., 1998. Speech Processing Using Automatically Derived Segmental Units: Applications to Very Low Rate Coding and Speaker Verification, in: In Proc. of Workshop on Text Speech Dialgue (TSD’98). Brno, Czech Republic, pp. 183–188.

Ariya, R., Sethy, A., Bhuvana, R., 2009. A new method for OOV detection using hybrid word/fragment system, in: In Acoustics, Speech and Signal Processing, ICASSP 2009. IEEE International Conference On. IEEE, pp. 3953–3956.

Avdagic, Z., Nuhic, A., Konjicija, S., 2007. Phoneme Recognition as a Member of Predefined Class using Hybrid Cascaded LVQ/Elman Neural Network, in: 2007 IEEE International Conference on Signal Processing and Communications. IEEE, pp. 1195–1198.

Awadalla, M., Abou_Chadi, F., Soliman, H., 2005. Development of an Arabic speech database, in: In Information and Communications Technology,Enabling Technologies for the New Knowledge Society. ITI 3rd International Conference on, pp. 89–100.

Awais, M., Masud, S., Shamail, S., 2006. Continuous Arabic Speech Segmentation using FFT Spectrogram, in: Innovations in Information Technology. IEEE, pp. 1–6.

Awais, M., Masud, S., Shamail, S., Akhtar, J., 2004. A hybrid multi-layered speaker independent Arabic phoneme identification system, in: In Intelligent Data Engineering and Automated Learning–IDEAL 2004. Springer, Berlin Heidelberg, pp. 416–423.

Azmi, M.M., Tolba, H., 2008. Syllable-based automatic Arabic speech recognition in different conditions of noise, in: 2008 9th International Conference on Signal Processing. IEEE, pp. 601–604.

Bacchiani, M., 1999. 1-153 "Speech recognition system design based on automatically derived units", PhD Thesis, Boston University.

Baghdasaryan, A.G., Beex, A.A., 2011. Automatic phoneme recognition with Segmental Hidden Markov Models, in: 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR). IEEE, pp. 569–574.

Bahl, L.R., Bellegarda, J.R., De Souza, P.V., Gopalakrishnan, P.S., Nahamoo, D., Picheny, M. a., 1993. Multonic Markov word models for large vocabulary continuous speech recognition. IEEE Transactions on Speech and Audio Processing 1, 334–344.

Baker, J., 1975. Stochastic modeling for automatic speech understanding, in: Readings in Speech Recognition. pp. 521–542.

Barroso, N., López de Ipiña, K., Hernández, C., Ezeiza, A., Graña, M., 2011. Experiments for the selection of sub-word units in the Basque context for semantic tasks. International Journal of Speech Technology 15, 49–56.

Baum, L., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. The annals of mathematical statistics 41, 164–171.

Bayeh, R., Lin, S., Chollet, G., Mokbel, C., 2004. Towards multilingual speech recognition using data driven source/target acoustical units association, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. I–521–4.

Biadsy, F., Hirschberg, J., Habash, N., 2009. Spoken Arabic dialect identification using phonotactic modeling, in: Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages. Association for Computational Linguistics. pp. 53–61.

Choueiter, G.F., 2009. "Linguistically-Motivated Sub-word Modeling with Applications to Speech Recognition" PhD Thesis,.

Cosi, P., Frasconi, P., Gori, M., Lastrucci, L., Soda, G., 2000. Competitive radial basis functions training for phone classification. Neurocomputing 34, 117–129.

Creutz, M., Stolcke, A., Hirsimäki, T., Kurimo, M., Puurula, A., Pylkkönen, J., Siivola, V., Varjokallio, M., Arisoy, E., Saraçlar, M., 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. ACM Transactions on Speech and Language Processing 5, 1–29.

Damien, P., Wakim, N., Egea, M., 2009. Phoneme-viseme mapping for Modern, Classical Arabic language, in: 2009 International Conference on Advances in Computational Tools for Engineering Applications. IEEE, Zouk Mosbeh, Lebanon, pp. 547–552.

Diehl, F., Gales, M., 2012. Morphological decomposition in Arabic ASR systems. Computer Speech & Language 26, 229–243.

Dua, M., Aggarwal, R., Kadyan, V., Dua, S., 2012. Punjabi Automatic Speech Recognition Using HTK. IJCSI International Journal of computer science Issues 9, 359–364.

Duran, D., Schütze, H., Möbius, B., Walsh, M., 2011. A Computational Model of Unsupervised Speech Segmentation for Correspondence Learning. Research on Language and Computation 8, 133–168.

Elmahdy, M., Gruhn, R., Abdennadher, S., Minker, W., 2011. Rapid phonetic transcription using everyday life natural Chat Alphabet orthography for dialectal Arabic speech recognition, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4936–4939.

Elshafei, M., Al-Muhtaseb, H., Al-Ghamdi, M., 2002. Techniques for high quality Arabic speech synthesis. Information sciences 140, 255–267.

Erdogan, H., Buyuk, O., Oflazer, K., 2005. Incorporating language constraints in sub-word based speech recognition, in: IEEE Workshop on Automatic Speech Recognition and Understanding, 2005. IEEE, pp. 98–103.

Essa, E., Tolba, A., Elmougy, S., 2008a. Combined Classifier Based Arabic Speech Recognition, in: INFOS2008. Cairo, pp. 27–29.

Essa, E.M., Tolba, A.S., Elmougy, S., 2008b. A comparison of combined classifier architectures for Arabic Speech Recognition, in: 2008 International Conference on Computer Engineering & Systems. IEEE, Cairo, pp. 149–153.

Forrest, K., Weismer, G., Milenkovic, P., Dougall, R.., 1988. Statistical Analysis Of Word Initial Voiceless Obstruents. Preliminary Data. Journal Of The Acoustical Society Of America 84, 115–23.

Fukada, T., Bacchiani, M., Paliwal, K.K., Sagisaka, Y., 1996. Speech recognition based on acoustically derived segment units, in: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96. IEEE, pp. 1077–1080.

Furuichi, C., Aizawa, K., Inoue, K., 2000. Speech recognition using stochastic phonemic segment model based on phoneme segmentation. Systems and Computers in Japan 31, 89–98.

Gemmeke, J., Bosch, L. ten, Boves, L., Cranen, B., 2009. Using sparse representations for exemplar based continuous digit recognition, in: Proc. EUSIPCO. pp. 24–28.

Ghrgen, F., Alpaydin, R., Gnlhakin, U., Alpaydin, E., 1994. Distributed and Local Neural Classifiers for Phoneme Recognition†. Pattern Recognition Letters 15, 1111–1118.

Hatazaki, K., Komori, Y., Kawabata, T., Shikano, K., 1989. Phoneme segmentation using spectrogram reading knowledge, in: International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. 393–396.

Hay, J., Bauer, L., 2007. Phoneme inventory size and population size. Language 83, 388–400.

He, X., Deng, L., 2008. Discriminative Learning for Speech Recognition: Theory and Practice. Synthesis Lectures on Speech and Audio Processing 4, 1–112.

Heintz, I., 2010. "Arabic language modeling with stem-derived morphemes for automatic speech recognition", PhD thesis, Ohio State University.

Hirsimäki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S., Pylkkönen, J., 2006. Unlimited vocabulary speech recognition with morph language models applied to Finnish. Computer Speech & Language 20, 515–541.

Huang, X., Acero, A., Hsiao-Wuen, H., 2001. Spoken language processing, First. ed. Prentice-Hall, Inc, New Jersey.

Huijbregts, M., McLaren, M., Van Leeuwen, D., 2011. Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4436–4439.

Iqbal, H., Awais, M., Masud, S., Shamail, S., 2008. On vowels segmentation and identification using formant transitions in continuous recitation of quranic arabic, in: New Challenges in Applied Intelligence Technologies. pp. 155–162.

Iwami, K., Fujii, Y., Yamamoto, K., Nakagawa, S., 2010. Out-of-vocabulary term detection by n-gram array with distance from continuous syllable recognition results, in: 2010 IEEE Spoken Language Technology Workshop. IEEE, pp. 212–217.

Jiang, L., Huang, X., 2003. Confidence measures using sub-word-dependent weighting of sub-word confidence scores for robust speech recognition. US Patent 6,539,353.

Juang, B., Rabiner, L., 2004. Automatic speech recognition–a brief history of the technology development. Georgia Institute of Technology. Atlanta.

Jurafsky, D., Martin, J., 2000. Speech & Language Processing, 1st ed. Pearson Prentice Hall.

KACST [WWW Document], 2012. . URL kacst.edu.sa

Kamakshi Prasad, V., Nagarajan, T., Murthy, H. a, 2004. Automatic segmentation of continuous speech using minimum phase group delay functions. Speech Communication 42, 429–446.

Kannan, A., Ostendorf, M., 1998. A comparison of constrained trajectory segment models for large vocabulary speech recognition. IEEE Transactions on Speech and Audio Processing 6, 303–306.

Kenney, F., 2008. "Automatic determination of sub-word units for automatic speech recognition", PhD Thesis, University of Edinburgh.

Kessens, J.M., Wester, M., Strik, H., 1999. Improving the performance of a dutchscr by modeling within-word and cross-word pronunciation variation. Speech Communication 29, 193–207.

Kim, D., Chan, H., Evermann, G., Gales, F., Mrva, D., Sim, C., 2005. Development of the CU-HTK 2004 broadcast news transcription systems, in: Proc. ICASSP. pp. 861– 864.

Kohonen, T., 1988. Self-organization and associative memory. springer series in information sciences 8, 312.

Kondo, K., Kamata, H., Ishida, Y., 1994. Speaker-independent spoken digits recognition using LVQ, in: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94). IEEE, pp. 4448–4451.

Kurimo, M., 1997. Training mixture density HMMs with SOM and LVQ. Computer Speech and Language 11, 321–343.

Kvale, K., 1993. Segmentation and labelling of speech.

Lamere, P., Kwok, P., Walker, W., Gouvea, W., Singh, E., Raj, R., Wolf, P., 2003. Design of the CMU Sphinx-4 decoder, in: Proceedings of the 8th European Conference on Speech Communication and Technology. pp. 1181–1184.

Lee, C., Juang, B., Soong, F.K., Rabiner, L.R., 1989. Word recognition using whole word and subword models, in: International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. 683–686.

Lee, K.F., 1988. Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System.

Lee, K.-F., Hon, H.-W., 1989. Speaker-independent phone recognition using hidden Markov models. IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 1641–1648.

Levinson, S., 1986. A unified theory of composite pattern analysis for automatic speech recognition, Computer speech processing. Prentice-Hall International (UK) Ltd, London.

Liang, M., Lyu, R., Chiang, Y., 2006. Using Speech Recognition Technique for Constructing a Phonetically Transcribed Taiwanese (Min-nan) Text Corpus, in: Ninth International Conference on Spoken Language Processing. pp. 1–12.

Ljolje, a., Riley, M.D., 1991. Automatic segmentation and labeling of speech, in: [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. 473–476 vol.1.

Lloyd, S., 1982. Least squares quantization in PCM. IEEE Transactions on Information Theory 28, 129–137.

Ma, D.-D., Zeng, X.-Q., 2012. An improved VQ based algorithm for recognizing speaker-independent isolated words, in: 2012 International Conference on Machine Learning and Cybernetics. IEEE, pp. 792–796.

Maalyl, I., Ahmed, A., 2002. New parameters for resolving acoustic confusability between Arabic phonemes in a phonetic HMM recognition system. Ashurst Lodge : WIT Press 1, 1–85312.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, in: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkely, University of California Press, California, pp. 281–297.

Majewski, P., 2008. Syllable based language model for large vocabulary continuous speech recognition of polish, in: Text, Speech and Dialogue. Springer, Berlin Heidelberg., pp. 397–401.

Mäntysalo, J., Torkkola, K., Kohonen, T., 1994. Mapping content dependent acoustic information into context independent form by LVQ. Speech communication 14, 119–130.

McDermott, E., Katagiri, S., 1991. LVQ-based shift-tolerant phoneme recognition. IEEE Transactions on Signal Processing 39, 1398–1411.

Mitchell, C., Setlur, A., Sukkar, R., 2003. Method and apparatus for recognition-based barge-in detection in the context of subword-based automatic speech recognition. US Patent 6,574,595.

MITCogNet, 2010. MITCogNet [WWW Document]. URL http://mitpdev.mit.edu/library/erefs/arbib/images/figures/A248_fig001.gif.

Monica, T., Nagarajan, T., 2011. Segmentation of speech signal into phonemes using two-level GMM tokenization, in: 2011 International Conference on Recent Trends in Information Technology (ICRTIT). IEEE, pp. 843–847.

Moreau, N., 2002. Htk-basic-tutorial. pdf [WWW Document]. Microsoft Corporation, Cambridge University …. URL http://www.info2.uqam.ca/~boukadoum_m/DIC9315/Notes/Markov/HTK_basic_tutorial.pdf.

Nahar, K.M.., Al-Khatib, W.G., Elshafei, M., Al-Muhtaseb, H., Alghamdi, M.M., 2012. Statistical Analysis of Arabic Phonemes for Continuous Arabic Speech Recognition. International Journal of Computer and Information Technology 1, 49–61.

Northwestern University, 2009. http://users.eecs.northwestern.edu/~wkliao/Kmeans/ [WWW Document]. Wei-keng Liao.

Novotney, S., Schwartz, R., Khudanpur, S., 2011. Unsupervised Arabic dialect adaptation with self-training, in: InterSpeech’11. pp. 1–4.

Paliwal, K., 1990. Lexicon-building methods for an acoustic sub-word based speech recognizer, in: International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. 729–732.

Pan, Y., Chang, H., Lee, L., 2007. Subword-based position specific posterior lattices (S-PSPL) for indexing speech information, in: Proc. Interspeech. pp. 318–321.

Parada, M., 2011. 182 Learning sub-word units and exploiting contextual information for open vocabulary speech recognition.

Parhi, K., 2002. Viterbi decoder architecture for interleaved convolutional code, in: Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002. IEEE, pp. 1934–1937.

Pellegrini, T., Lamel, L., 2009. Automatic Word Decompounding for ASR in a Morphologically Rich Language: Application to Amharic. IEEE Transactions on Audio, Speech, and Language Processing 17, 863–873.

Petek, B., Andersen, O., Dalsgaard, P., 1996. On the robust automatic segmentation of spontaneous speech, in: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP ’96. IEEE, pp. 913–916.

Pieraccini, R., Rosenberg, A.E., 1989. Automatic generation of phonetic units for continuous speech recognition, in: International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. 623–626.

Prasad, T., Kohli, M., 2010. Vector Quantization of Microarray Gene Expression Data, in: Proceedings of the World Congress on Engineering 2010 Vol I WCE. LONDON, UK, p. 5.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286.

Rahman, M., Khan, M., Bhuiyan, M., 2012. Continuous Bangla Speech Segmentation, Classification and Feature Extraction. nternational Journal of Computer Science 9, 67–75.

Raju, M., 2003. Automatic Language Identification based on Acoustic Sub-word units. Master's thesis, Dept. ECE, National Institute of Technology, suratkal.

Rastrow, A., Sethy, A., Ramabhadran, B., Jelinek, F., 2009. Towards using hybrid, word, and fragment units for vocabulary independent LVCSR systems, in: Proc. Interspeech. pp. 1931–1934.

Riley, M., Ljolje, A., 1996. Automatic generation of detailed pronunciation lexicons, in: Automatic Speech and Speaker Recognition. Kluwer Acadnic Publisher, pp. 285–301.

Rotovnik, T., Maučec, M.S., Kačič, Z., 2007. Large vocabulary continuous speech recognition of an inflected language using stems and endings. Speech Communication 49, 437–452.

Sadohara, K., Lee, S., Kojima, H., 2006. DOMAIN-INDEPENDENT TOPIC SEGMENTATION USING A STRING KERNEL ON RECOGNIZED SUB-WORD SEQUENCES, in: 2006 IEEE Spoken Language Technology Workshop. IEEE, pp. 30–33.

SaiJayram, A.K. V., Ramasubramanian, V., Sreenivas, T. V., 2002. Robust parameters for automatic segmentation of speech, in: IEEE International Conference on Acoustics Speech and Signal Processing. IEEE, pp. I–513–I–516.

Sarada, G.L., Lakshmi, A., Murthy, H.A., Nagarajan., T., 2009. Automatic transcription of continuous speech into syllable-like units for Indian languages. Sadhana 34, 221–233.

Schlüter, R., Gollan, C., Hahn, S., 2008. Development of Large Vocabulary ASR Systems for Mandarin and Arabic. Voice Communication (SprachKommunikation), 2008 ITG Conference on , vol., no., pp.1,4, 8-10 Oct. 2008 1, 8–10.

Schmied, J., 2008. CHEMNITZ UNIVERSITY OF TECHNOLOGY, in: English Projects in Teaching and Research in Central Europe: Proceedings of the Freiberg Conference. Cuvillier Verlag, p. 1.

Sekhar, C., Lee, W., Takeda, K., Itakura., F., 2003. Acoustic modeling of subword units using support vector machines, in: Workshop on Spoken Language Processing. India, pp. 79–86.

Selouani, S., Caelen, J., 1999. A hybrid learning vector quantization/time-delay neural networks system for the recognition of arabic speech, in: Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99). Bogazici Univ, Instanbul, Turkey, pp. 709–713.

Sharma, A., Perala, S., Darshni, P., 2012. Objects Control through Speech Recognition Using LabVIEW. International Journal of Electronics and computer science engineering 2, 102–106.

Sharma, M., Mammone, R., 1996. "Blind" speech segmentation: automatic segmentation of speech without linguistic knowledge, in: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96. IEEE, pp. 1237–1240.

Singh, P., Lehal, G., 2010. Statistical syllables selection approach for the preparation of Punjabi speech database, in: International Conference for Internet Technology and Secured Transactions (ICITST), 2010. IEEE, pp. 1–4, 8–11.

Singh, R., Raj, B., Stern, R., 2002. Automatic generation of subword units for speech recognition systems. IEEE Transactions on Speech and Audio Processing 10, 89–99.

Sitaram, R., Sreenivas, T., 1994. Phoneme recognition in continuous speech using large inhomogeneous hidden Markov models, in: Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, p. I/41–I/44.

Sofya, F., Al-Obadi, M., 2008. Syllabic Segmentation Algorithm of Arabic Word. uomosul.edu.iq 1, 21–31.

Soltau, H., Saon, G., Kingsbury, B., Kuo, J., Mangu, L., Povey, D., Zweig, G., 2007. The IBM 2006 Gale Arabic ASR System, in: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07. IEEE, pp. IV–349–IV–352.

Sreenivas, V., 2004. Segmental Ã-means Training of an Ergodic-HMM of Sub-word HMMs for Language Identification.

Steinhaus, H., 1956. Sur la division des corp materiels en parties. Bull. Acad. Polon. Sci 4(12), 801–804.

Svendsen, T., Paliwal, K., Harborg, E., Husoy, P.O., 1989. An improved sub-word based speech recognizer, in: International Conference on Acoustics, Speech, and Signal Processing. IEEE, pp. 108–111.

Szoke, I., Burget, L., Cernocky, J., Fapso, M., 2008. Sub-word modeling of out of vocabulary words in spoken term detection, in: 2008 IEEE Spoken Language Technology Workshop. IEEE, pp. 273–276.

Tak, G., Bhargava, V., 2010. Clustering Approach in Speech Phoneme Recognition Based on Statistical Analysis. Recent Trends in Network Security and Applications 89, 483–489.

Tejedor, J., Wang, D., Frankel, J., King, S., Colás, J., 2008. A comparison of grapheme and phoneme-based units for Spanish spoken term detection. Speech Communication 50, 980–991.

Thangarajan, R., Natarajan, a. M., Selvam, M., 2009. Syllable modeling in continuous speech recognition for Tamil language. International Journal of Speech Technology 12, 47–57.

Tolba, M., Nazmy, T., Abdelhamid, A.A., Gadallah, M., 2005. A novel method for Arabic consonant/vowel segmentation using wavelet transform. International Journal on Intelligent Cooperative Information Systems, IJICIS 5, 353–364.

Venkatesh, N., Gulati, R., Bhujade, R., Chandra, M.G., 2011. Fixed-point implementation of isolated sub-word level speech recognition using hidden Markov models, in: Proceedings of the 2011 ACM Symposium on Applied Computing - SAC '11. ACM Press, New York, New York, USA, p. 368.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K.J., 1989. Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 328–339.

Walliczek, M., Kraft, F., Jou, S.-C., Schultz, T., Waibel, A., 2006. Sub-word unit based non-audible speech recognition using surface electromyography, in: Proc. Interspeech, Pittsburgh, PA (2006). pp. 1487–1490.

Walpole, R., Myers, R., Myers, S., Ye, K., 2007. Probability and statistics for engineers and scientists. Pearson Prentice Hall, Boston : Upper Saddle River, NJ.

Westendorf, C., Jelitto, J., 1996. Learning pronunciation dictionary from speech data, in: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96. IEEE, pp. 1045–1048.

Wikipedia, 2013. K-means clustering [WWW Document]. URL http://en.wikipedia.org/wiki/K-means_clustering#History

Xie, L., Yang, Y., Liu, Z., 2011. On the effectiveness of subwords for lexical cohesion based story segmentation of Chinese broadcast news. Information Sciences 181, 2873–2891.

Yokota, M., Katagiri, S., McDermott, E., 1988. Learning in an LVQ Based Phoneme Recognition System, 7E/CE Technical Report, SP88-104. Jpn.

Young, S., Evermann, G., Gales, M., 2006. The HTK book version 3.4. University Of Cambridge, UK.

Yu, S., Xu, C., Liu, H., Chen, Y., 2011. Statistical analysis of Chinese phonemic contrast. Phonetica 68, 201–14.

Zhang, J., Nakamura, S., 2006. Automatic Derivation of a Phoneme Set with Tone Information for Chinese Speech Recognition Based on Mutual Information Criterion, in: 2006 IEEE International Conference on

Acoustics Speed and Signal Processing Proceedings. IEEE, pp. I–337–I–340.

Zhang, Y., 2009. "Unsupervised spoken keyword spotting and learning of acoustically meaningful units",PhD, Massachusetts Institute of Technology.

Zue, V., Lamel, L., 1986. An expert spectrogram reader: A knowledge-based approach to speech recognition, in: ICASSP ’86. IEEE International Conference on Acoustics, Speech, and Signal Processing. Institute of Electrical and Electronics Engineers, TOKYO, pp. 1197–1200.

VITAE

Personal Information

- **Name:** Khalid Mohamed Oqlah Nahar.
- **Birth Place**: KufrKhal-Jarash.
- **Birth Date:** October 3, 1970.
- **Nationality:** Jordanian.
- **Marital Status:** Married.
- **No. of Children**: 5.
- **Mail-Address**: Computer Science Department
  Faculty of Information Technology
  Yarmouk University
  Irbid-Jordan
  - **Office Tel**: 00962-(0)2-7211111 Ext. 2535
  - **Home Tel**: 00962-(0)2-7211111 Ext. 3350
  - **Mobile-Jordan**: 0777283624.
  - **Mobile-Saudi Arabia**: 0507508624
  - **E-mail address/Yarmouk: khalids@yu.edu.jo**
  - **E-mail address/KFUPM: nahar@kfupm.edu.sa**
  - **E-mail address/Yahoo: nawasreh@yahoo.com**

EDUCATION

Degree	University	Graduation Date	Cumulative Average	Grade
BSc In Computer Science	Yarmouk University / Irbid-Jordan	19-1-1992	79.0	Very Good
MSc In Computer Science	Yarmouk University / Irbid-Jordan	14-6-2005	91.9	Excellent
PhD In Computer science and Engineering	King Fahd University Of Petroleum and Minerals	15-5-2013	3.8/4	Excellent

SPOKEN LANGUAGES

	Arabic	English
Oral	**Excellent**	**Very good**
Writing skills	**Excellent**	**Very good**

PROGRAMMING LANGUAGES AND TOOLS

- VP .NET
- Visual C++ 6.0, C#,C.
- Oracle SQL, PL/SQL
- Turbo Prolog, Pascal from Ver 3 up Ver 9, Top-Speed Modula 2,Fortran 77
- CLIPS
- MATLAB 2010a (professional)
- Sphinx ASR system
- HTK ASR system
- Pocket-Sphinx

WORKSHOPS AND TRAINING COURSES

- Introduction to Operating System (UNIX).

- HP-UNIX System Administration (50 hours).
- ECDL (Microsoft certificate).
- First and Second Student Conferences (SSC1 and SSC2) in Saudi-Arabia
- Presenting a paper in the graduate day-competion at KFUPM
- MATLAB professional Programming

HISTORY OF PROFESSIONAL EMPLOYMENT

No.	Period From To		Job	Organization
1	18/2/92	8/6/92	Teacher	Directorate of Education
2	26/9/92	10/4/93	Instructor	Ibn-Khaldoun College-Irbid
3	10/4/93	10/4/2003	Teaching Research Assistant	Yarmouk University
4	1/6/2005	Now A Leave Without Pay	Instructor	Yarmouk University
5	1/9/2010	Now	PhD student/Lec-B	King Fahd University Of Petroleum and Minerals

- I have taught several courses At Yarmouk University:
  - CS100: An introduction to computer and its applications (Non-computer science students Office+SharePoint).
  - CS101: An introduction to programming (C++ programming language).
  - CS102: An introduction to computer science.
  - CS114: AI programming (prolog language).
  - CS117: Object oriented programming.
  - CS250: Data structures with Object Oriented Programming.
  - CS333: Data communication and networking.
  - CS 416: Computer Graphics With OpenGl.

RESEARCH INTEREST

- Continuous Speech Recognition
- Nero-Processing systems and algorithms (ANN, LVQ, HMM)
- Artificial Intelligence (AI).
- Natural Language Processing (NLP) (Arabic Processing).
- Information Retrieval.
- Data Security and Encryption.
-

PUBLICATIONS

- ✓ Al-Kabi, Mohammed; Kanaan, Ghassan; Al-Shalabi, Riyad and **Nahar, Kahlid**. Bani-Ismail, Basel Mohammed "Statistical Classifier of the Holy Quran Verses (Fatiha And Yaseen Chapters)", Journal of Applied Sciences 5(3): 580-583, 2005.
- ✓ Al-Shalabi, Riyad, Kanaan, Ghassan, **Nahar, Kahlid** and Madalal, Mohammad. "Question Answering System For Arabic Language Using N-gram Technique", International Journal of Applied Science and Computations Vol. 12, N0.2 August 2005, pages 113 - 128.
- ✓ **Khalid Nahar**, Osama Abu Abbas and Mohammad Tubishat. HRO encryption system. In the proceedings of the 8th international Arab Conference on Information Technology (ACIT2007). November 26-28, 2007: 73-79.
- ✓ Osama Abu Abbas, **Khalid Nahar** and Mohammad Tubishat. ARAE cipher system. In the proceedings of the 8th international Arab Conference on Information Technology (ACIT2007). November 26-28, 2007:90-93.
- ✓ Mohammad Tubishat, **Khalid Nahar** and Osama Abu Abbas * 1,"Dual Language Encryption System", ABHATH AL-YARMOUK: "Basic Sci. & Eng." Vol. 18, No. 2, 2009, pp. 215- 226 Received on March 10, 2009 Accepted for publication on June 1,
- ✓ **Khalid Nahar**, Osama Abu Abbas and Ahmad Mansour,"Hybrid Cipher System" , ABHATH AL-YARMOUK: "Basic Sci. & Eng." Vol. 19, No. 1, 2010. Coming in 2010.
- ✓ **Khalid Nahar** and Izzat alsmadi "The Automatic Grading for online exams with Essay Questions Using Statistical and Computational Linguistics Techniques", MASAUM Journal of computing, Volume 1 Issue 2, September 2009.
- ✓ **Khalid M. O Nahar**, Mustafa Elshafei, Wasfi G. Al-Khatib, Husni Al-Muhtaseb, Mansour M. Alghamdi. "*Statistical Analysis of Arabic Phonemes Used in Arabic Speech Recognition*", 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part I, pp 533-542.
- ✓ **Khalid M. O Nahar**, Mustafa Elshafei, Wasfi G. Al-Khatib, Husni Al-Muhtaseb, Mansour M. Alghamdi. "*Statistical Analysis of Arabic Phonemes for continuous Speech Recognition*", International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 01– Issue 02, November 2012.
- ✓ **Nahar, Khalid M.O**; Al-Khatib, Wasfi G.; Elshafei, Moustafa; Al-Muhtaseb, Husni; Alghamdi, Mansour M., "Data-driven Arabic phoneme recognition using varying number of HMM states," Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on , vol., no., pp.1,6, 12-14 Feb. 2013 doi: 10.1109/ICCSPA.2013.6487258.

PRIZES

- Wining The 5th position for presenting a paper in the second student conference (SSC2).
- The first-top student Certificate in my Master-Degree at Yarmouk University-Jordan.